Empirical Software Engineering (2021) 26: 80
https://doi.org/10.1007/s10664-021-09972-6

Reliability of self-rated experience and confidence as predictors for students' performance in software engineering

Results from multiple controlled experiments on model comprehension with graduate and undergraduate students

Marian Daun[1] · Jennifer Brings[1] · Patricia Aluko Obe[1] · Viktoria Stenkova[1]

Accepted: 14 April 2021 / Published online: 9 June 2021

Abstract
Students' experience is used in empirical software engineering research as well as in software engineering education to group students in either homogeneous or heterogeneous groups. To do so, students are commonly asked to self-rate their experience, as self-rated experience has been shown to be a good predictor for performance in programming tasks. Another experience-related measurement is participants' confidence (i.e., how confident is the person that their given answer is correct). Hence, self-rated experience and confidence are used as selector or control variables throughout empirical software engineering research and software engineering education. In this paper, we analyze data from several student experiments conducted in the past years to investigate whether self-rated experience and confidence are also good predictors for students' performance in model comprehension tasks. Our results show that while students can somewhat assess the correctness of a particular answer to one concrete question regarding a conceptual model (i.e., their confidence), their overall self-rated experience does not correlate with their actual performance. Hence, the use of the commonly used measurement of self-rated experience as a selector or control variable must be considered unreliable for model comprehension tasks.

Keywords Student performance · Self-rated Experience · Confidence · Model comprehension · Conceptual models

1 Introduction

Experience levels are important predictors, not only in software engineering education but also in empirical studies. In an educational setting, experience is often used to distribute

Communicated by: Scott Fleming

✉ Marian Daun
marian.daun@paluno.uni-due.de

Extended author information available on the last page of the article.

students fairly (e.g., Marshall et al. (2016); Katira et al. (2004)) or to ensure that weak students receive more attention and support (e.g., Webb et al. (1998); Jensen (2015)). In empirical studies participants' experience often serves as a covariate, e.g., to validate certain assumptions (cf. Wieringa (2010); Goodwin and Goodwin (2016)) or to evaluate particular threats (e.g., Wohlin et al. (2000); Robson (2016); Campbell and Stanley (1963); Cook and Campbell (1979)). Additionally, participants' experience is often used to ensure that the field of participants is homogeneous regarding their experience (cf. Kitchenham et al. (2002); Sjøberg et al. (2003); Fucci et al. (2015)), or as a selection operator for distinct treatment and control groups (cf. Sjøberg et al. (2003); Sjøberg DIK et al. (2002); Kitchenham et al. (2002); Wieringa (2010)).

Thus, two questions arise (a) how can experience be measured, and (b) is this experience measure a reliable predictor for performance. In this sense, we call a predictor reliable or good if the measurement correlates with the predicted variable (i.e., if a measured low experience predicts low performance and a measured high experience predicts high performance).

Even though experience is commonly used as a predictor for participants' and students' performance, experience itself is hard to measure objectively. Therefore, various ways to measure experience have been subjects of investigations. Commonly used measurements for experience include duration like *experience in years* or *experience in months* (e.g., Pinto et al. (2019); Sillito et al. (2008)) or educational levels like *bachelor degree* or *master degree* (e.g., Feigenspan et al. (2012)). While these types of measurements are easy to determine and seem objective, they can also be quite unreliable. For instance, *experience in years* as measurement for experience does not consider the intensity of work done in these years. Similarly, one year of doing something intensively can provide more experience than doing the same thing rarely over several years. Therefore, another popular way to measure experience is to rely on self-rated experience on some kind of scale (e.g., Nugroho (2009)). This kind of measurement faces the problem that equally experienced persons might rate their experience differently (e.g., due to different personality traits). Nevertheless, studies found self-rated experience to be a good measure for experience and a good predictor for performance (cf. Bergersen et al. (2011); Höst et al. (2000)). This seems particularly true for programming tasks (cf. Feigenspan et al. (2012)). Bergersen et al. (2014) report on multiple experience measurements that relate to programming skills and conclude that those can be used as predictors for performance where a medium or large sized correlation is present.

In addition, confidence as a self-rated measurement is also used as a predictor for performance (cf. Morgan and Cleave-Hogg (2002)). Confidence measures a person's faith in their abilities to solve a certain task. In doing so, confidence can be seen as a measurement that is related to experience, as experience can be seen as a person's source of confidence in their abilities (cf. Jørgensen et al. (2004)). Therefore, we investigate experience on a personal level and confidence on a task level. However, there exist also measurements of confidence on an abstract capability level. For instance, Layman et al. (2005) show that self-confidence of students in their programming skills indeed correlates with their performance, i.e., very self-confident students perform better than less self-confident students.

Consequently, self-rated experience and confidence are commonly used as predictors for performance in software engineering tasks. However, in contrast to the results of studies from programming tasks, several studies we conducted over the past four years in the context of model-based engineering indicate that self-rated experience might not always be a good predictor. Hence, the question is whether self-rated experience and confidence are good predictors when it comes to model comprehension. To answer this question, this paper contributes an observational study investigating whether self-rated experience and confidence are good predictors for students' performance in model comprehension tasks. In this study, we investigate results obtained from several experiments that compare students' confidence in giving a correct answer, performance, and self-rated experience. Our results show that while students seem to be sometimes able to assess if an answer they have just given is correct or not via their self-rated confidence, there is no significant correlation between their performance and their self-rated experience. Consequently, as far as teaching and conducting empirical studies in model-based engineering is concerned, students' self-rated experience cannot be used as a reliable predictor for their performance.

This paper is structured as follows: Section 2 discusses related studies in the field. Based on the related work, Section 3 defines the basic terminology for this paper: experience, confidence, and performance. Section 4 discusses the context of our investigation (i.e., previous experiments used as data sources). Section 5 gives insight into our experimental setup. Section 6 reports the study's results and Section 7 discusses our findings. Section 8 concludes the paper.

2 Related Work

Students' performance has been under investigation for empirical research as well as for educational purposes. Related works specifically deal with predicting and measuring students' performance and the use of the predicted performance to group students either in homogeneous or heterogeneous groups. In this section, we will briefly summarize related work in empirical software engineering research (Section 2.1) and in software engineering education (Section 2.2). Finally, we will give an overview of related work on prediction models for students' performance also considering other research disciplines in Section 2.3.

2.1 Students' Performance in Empirical Software Engineering Research

Students' performance is commonly discussed in empirical software engineering. The main focus of investigation in doing so is to answer the question whether students' and professionals' performance differs. If students' and professionals' performance were comparable, student experiments would be generalizable to industrial practice. For instance, Höst et al. (2000) show for smaller tasks of judgment that there are only minor differences in performance between software engineering students and professionals. Runeson (2003) investigated the differences between freshmen, graduates, and professionals, finding that freshmen's performance significantly differs from graduates' and professionals' performance. In line with this finding, Tichy (2000) concludes that graduate students' performance is close to professional status. A survey by Salman et al. (2015) compared students with professionals in the context of a test-driven development experiment. Some differences in the quality of code were found, but students performed not significantly worse (or better) than professionals. However, please note that also a plethora of contradictory works exists that suggests that students' performance is not generalizable to industry professionals performance per se (cf. e.g., Berander (2004); Feldt et al. (2018)).

Another aspect of students' performance in empirical research deals with the use of students' predicted performance as a selector for treatment and control groups (cf. Kitchenham et al. (2002); Mkpojiogu and Hussain (2017); Raza et al. (2017); Sjøberg et al. (2003); Sjøberg DIK et al. (2002); Wieringa (2010)). Alternatively, the participants' experience, their self-perception, and estimated performance are controlled as covariates (cf. Goodwin and Goodwin (2016); Wieringa (2010)). However, if participants have a bad self-perception, covariates might not indicate a significant interaction effect despite participants' actual performance. Hence, Feigenspan et al. (2012) investigated whether different measurements for experience are good predictors for students' actual performance in programming tasks. They investigated years of experience, education, self-estimation, and size of programs written. The findings show that self-rated experience on a semantic differential scale can be used to predict performance for programming tasks.

Briefly summarized, related research is centered on the question whether the performance of students is comparable to the performance of industry professionals. While the results are not conclusive, there are indications that under certain conditions this is the case, for instance, for graduate students. However, it is recognized that the experience of participants seems to play an important role (i.e., that students are generalizable to a professional level if they are experienced). Thus, this indicates that experience might be a useful predictor for performance.

2.2 Students' Performance in Software Engineering Education

In software engineering education, on the one hand student experiments are used to provide insight into industrial practice (cf. Carver et al. (2003)). Thus, much of the findings discussed for empirical software engineering also hold true for software engineering education. On the other hand, estimated student performance is used in teaching approaches to group students in such a way that students' actual performance is increased (e.g., Webb et al. (1998)).

For instance, Jensen (2015) investigated how to increase students' performance using team work in projects. It showed that particularly reluctant students' teamwork skills benefit from consultation in teams that also consist of high performing students. In addition, Zhang et al. (2014) studied software engineering students and found that increased knowledge of teamwork skills did not translate into better practice skills.

Student performance as a selector for student grouping is not specific to the software engineering field. For instance, Cen et al. (2015) created a method to reach optimal grouping for optimal group performance. A study of this method was conducted among 122 engineering and molecular biology students, it showed that students' performance in teams which have been assigned to achieve heterogeneous teams was better than in self-assigned teams.

We can conclude that in software engineering education a need for predicting students' performance exists. Therefore, one feasible approach seems to be the use of students' experience. In this paper, we will more closely investigate whether self-rated experience can be used to predict performance in model comprehension tasks.

2.3 Prediction Models for Students' Performance

Related work on prediction models for students' performance has been done in non-computer science research as well as in computer science research on programming. In addition, general investigations exist.

2.3.1 Prediction of Students' Performance in Programming

Literature shows that common predictors for students' performance in programming are the years of experience (e.g., Pinto et al. (2019); Sillito et al. (2008)), the education of the participants (e.g., graduates as participants with high experience and undergraduates as students with low experience (cf. Ricca et al. (2007))), self-rated experience measured on a 5-point semantic differential scale (e.g., Bunse (2006)), or the size of the most complex previous work (e.g., Müller (2004)). In other programming experiments students' experience was estimated by a pretest (e.g., Biffl (2003)) or prehoc defined by the supervisor (e.g., Hannay et al. (2010); Arisholm et al. (2007)). Feigenspan et al. (2012) conducted a study to compare the prediction capabilities of the different measurements and found out that for programming tasks self-rated students' experience on a semantic differential scale or in comparison to classmates is a good predictor for students' performance.

Hagan and Markham (2000) and Byrne and Lyons (2001) investigated influences on student performance in programming. Byrne and Lyons have shown that prior programming experience increases students' performance. Hagan and Markham reported that students with prior knowledge in at least one programming language performed better in an introductory programming course. Specifically, they concluded that the more programming languages a student knows, the better their performance.

Bergersen et al. (2011) analyzed two data sets, one with consultants and the other with students. The authors show that expertise can serve as a predictor for performance. Expertise was measured as the combination of extended experience, consensual agreement, self-assessment, reliability, and knowledge. Programming performance was measured as time and quality.

Layman et al. (2005) investigated that students who are in general more self-confident regarding their programming skills do perform better than less self-confident students in programming tasks.

In summary, many researchers found that experience and confidence can be used as predictors for students' performance in programming tasks. However, there exist different ways to measure experience and confidence that seem to be more or less suitable. An often valued measurement is the use of self-rated experience on a semantic differential scale as well as experience in terms of educational achievements. However, these findings were made for programming tasks. Thus, generalizability to other domains must be questioned. In this paper, we will examine if this also holds for model comprehension tasks.

2.3.2 Prediction of Students' Performance in Other Disciplines

Rex and Roth (1998) compared self-reported computer experience with computer self-efficacy, and performance. Experience was measured as years of computer experience, as the number of prior computer courses completed, and as current average hours per week of computer use. Performance was measured in course grades. Significant correlations exist between performance and experience in computer use and performance and computer self-efficacy.

Eskew and Faley (1988) report on their findings from a freshmen college level accounting course. Eskew and Faley investigated which measurements can serve as good predictors for students' performance. Among other aspects (e.g., effort, grade point average) they found experience to be a significant predictor of performance. In this setting, experience was measured by the amount of previous education in similar subjects (e.g., bookkeeping, math).

Morgan and Cleave-Hogg (2002) examined the relationship between experience, confidence, and performance for medical students. Confidence was measured on a 5-point semantic differential scale, experience as the number of times a certain situation had been encountered or skill had been performed, and performance as grades. As result Morgan and Cleave-Hogg report that there is a significant correlation between confidence and experience, but no correlations between confidence and performance, or experience and performance.

In summary, it can be concluded that experience can be used as a predictor for performance in different areas. However, studies also exist that contradict this finding and explicitly state that no correlation between experience and performance or confidence and performance are observable. Hence, it must be assumed that experience and confidence cannot be used as predictors in all cases. Therefore, this paper will investigate whether self-rated experience and performance can be used as predictors for students' performance in model comprehension tasks.

2.3.3 General Works on Students' Performance

Boud and Falchikov (1989) reviewed existing experiment reports to determine if students have the ability to correctly self-assess their performance. The authors' findings are inconclusive: While some primary studies report over-rating by students, others report under-rating. However, Boud and Falchikov found that self-assessment results of students' performance are better when a 5-point scale is used than when asking for a percentage value. In addition, Falchikov and Boud (1989) found that older students tend to become better at self-rating.

Mishra et al. (2014) present a method to predict students' performance based on their gender, parents' education, marks, and abilities. Data from 215 students were collected and analyzed with random decision trees and the J48 decision tree algorithm (Witten et al. (1999)). The result of the study shows that leadership and drive affect students' performance as well as performance in the previous semester. Other researchers more broadly investigated the impact of gender on performance and other benchmarks in a professional environment. For instance, James et al. (2017) found that most self-perceived differences in a professional environment are based on personality factors rather than on differences in gender. However, for self-perceived overall performance they found female participants to be significantly more satisfied with their perceived performance compared to male participants.

Thus, the related work on students' performance substantiates the assumptions made before. There exist different opinions regarding the influence of experience on performance. To some extent, this seems to depend on the concrete task to be solved. Therefore, this study investigates the influence of experience and confidence on performance in model comprehension tasks.

3 Background and Terminology

This paper reports an observational study to investigate whether experience and confidence can be seen as reliable predictors for the students' actual performance. The relationship between students' experience, confidence, and performance for model comprehension tasks is analyzed. Therefore, it is first necessary to understand what experience, confidence, and performance mean and why these measurements are important. Unfortunately, the current state of the art is missing clear definitions. In most scientific literature the terms are used colloquially and never precisely defined. Even though the ISO/IEC/IEEE International Standard 24764 (cf. ISO/IEC/IEEE (2010)) uses these terms regularly throughout the standard to define other terms, it does not define them clearly either. Consequently, we next define experience, confidence, and performance in the sense commonly used in literature and by the standard to define the context parameters of our empirical investigation.

3.1 Experience

The term experience is used for many different aspects. For this paper experience can best be defined as “practical knowledge, skill, or practice derived from direct observation of or participation in events or in a particular activity”[1] and “the length of such participation; i.e., has 10 years’ experience in the job”[1], which is in accordance with experience definitions from the education domain (cf. e.g., Kirschner (1992); Cushion et al. (2003)). In this sense, experience refers to knowledge as well as to skills the participants possess. As such, experience is hard to measure objectively.

As has been discussed in Section 2, a multitude of approaches exists to define measurements for experience with the aim that experience indeed explains the performance of people and is therefore a good predictor. The underlying assumption is that more experienced people perform better due to their experience (i.e., due to their craftsmanship and their knowledge in the respective area of expertise). Note that this is at the center of our investigation to find out whether experience indeed leads to better performance (for model comprehension tasks). However, as discussed, there is a need to define measurements for experience. We investigate two popular measurements:

– *Degree Program.* It is often acknowledged that graduate students are far more experienced than undergraduate students. Particularly, investigations exist regarding the generalizability of student participants to the level of practitioners. Many of these investigations conclude that graduate students are sufficiently experienced, to allow for this generalization while undergraduate students are not.
– *Self-rated Experience.*[2] The idea behind self-rated experience is that students can assess their own strengths and weaknesses; and when asked to self-rate their level of experience the result is a good proxy for their performance. Investigations have shown that this is often true (cf. Boud and Falchikov (1989)). Particularly, in the area of programming, studies have shown that the self-rated experience is indeed a good predictor for the students’ performance in programming and code comprehension tasks (cf. Feigenspan et al. (2012)).

3.2 Confidence

Confidence refers to a mixture of “a feeling or consciousness of one’s powers or of reliance on one’s circumstances”[1] and certitude, i.e., “the quality or state of being certain”[1]. Therefore, confidence is related to experience as it can be seen as an indicator for people having faith in their experience and that this experience leads them to making a correct decision. In recent software engineering research the focus of investigating confidence is often on

[1] Definition by https://www.merriam-webster.com

[2] Note that there exists a wide variety of experience measurements. Sometimes it is explicitly differentiated between skills and experience. For instance, Bergersen and Gustafsson (2011) define experience as a duration of doing something and skills as related to the ability of doing something. In this sense, our measurement of self-rated experience could also be considered a measurement of skills. However, we focus on the measurement of self-rated experience as predictor rather than on defining a precise measurement for experience.

comparing male and female participants. Studies show that female software engineers or software engineering students are often less self-confident regarding their performance than male ones although they do not perform worse. For instance, Bastarrica and Simmonds (2019) have shown that self-confidence cannot be used to predict performance for male and female groups. In contrast, Kumar (2008) has shown that this seems to hold true only for pre-surveys, not for post-surveys. Thus, this finding indicates that using self-confidence measured after conducting the task might indeed be a good predictor for performance regardless of the gender of the participants. However, the focus of our investigation lies more on certainty. For each individual question asked, we asked the participants how confident they are in their decision. Thus, for confidence we use the measurement:

- *Self-rated Confidence.* Much akin to self-rated experience, we investigate whether the self-rated confidence of students is a good predictor for the actual performance. While we measure self-rated experience on a personal level (i.e., how experienced is the participant), self-rated confidence is measured on a task level (i.e., how confident is the participant in one concrete given answer).

3.3 Performance

The use of the term performance in software engineering research is commonly derived from the technical performance of algorithms and machines, even when describing human activity. However, performance in this work does not refer to efficiency but to correctly carrying out tasks (i.e., how good are the students in the tasks they have been given). Note that this often also implies a timely component in the sense of efficiency, i.e., the best performing student is the student that solves all tasks correctly in the least amount of time. However, we are more interested in the correct task fulfillment. This is not only a prerequisite for the second part but also closely related to the commonly used definition of student performance in the educational literature (cf. e.g., Dick et al. (2001); Mcdowell et al. (2003); Newhall et al. (2014)). Student performance is commonly associated with passing an exam, the grades received, or the knowledge gained - mostly independent from the time used for this. Additionally, the use of qualitative performance measurement is complicated as quality cannot be associated to an unitary understanding (cf. Bergersen et al. (2014)). Hence, we define performance as

- *Average Correctness.* The participants are given questions to be answered. For each of these tasks it is determined whether the answer is correct or incorrect. Performance is now defined as the ratio of correct answers given compared to all answers. This means that the highest-performing students are those students who give only correct answers. This definition is also much akin to the performance definition for exams.

4 Context

In this study, we use data obtained from four experiments investigating the effects of different modeling alternatives for validation tasks. The experiments differ in the concrete modeling technique under investigation and thus in materials used and in concrete questions asked, but the principal procedure and tasks are the same for all experiments. All experiments deal with model perception tasks and thus allow for comparability w.r.t. manual validation tasks of models.

In the original experiments, as independent variable we typically investigated different model-based notation formats displaying a specification excerpt to be inspected by ad-hoc review. As dependent variables we typically determined:

- *Effectiveness:* percentage of correct decisions made.
- *Efficiency:* average time spent for making a correct decision.
- *User Confidence:* average confidence a participant claims in the correctness of their answers. Measured on a 5-point semantic differential scale, where 1 means very unconfident and 5 means very confident.
- *Subjective Supportiveness:* average result of self-rated standardized questionnaire items from the TAM 3 (Technology Acceptance Model v.3, cf. Venkatesh and Bala (2008)) for *perceived usefulness*, *perceived ease of use*, and *computer self-efficacy*.

Table 1 provides a brief overview of the different experiments. In the following, we briefly outline the intention of each experiment, provide the most important descriptive statistics and summarize the findings of the experiment.

Table 1 Overview of included experiments

Experiment	Goal of Study	Participants
Experiment 1: Representation of inconsistencies	Investigate whether the integrated representation (i.e. the merged diagram) is advantageous compared to the separate representation (i.e. the original bMSC diagrams) for reviews with respect to their effectiveness and efficiency.	68 graduate students
Experiment 2: Representation of recurring instances	Investigate whether the use of instance models showing different collaborative system network configurations is beneficial for the manual validation of collaborative system networks.	55 undergraduate students 45 graduate students
Experiment 3: Comparison review model vs functional design	Investigate whether reviewing the functional design after transformation into the format of MSC is advantageous, in particular with respect to effectiveness, efficiency, user confidence, and subjective supportiveness, compared to common reviews.	60 undergraduate students
Experiment 4: Comparison review model vs original specifications	Investigate whether reviewing a dedicated review model is more effective and efficient than reviewing behavioral requirements and functional design and whether the experience and skill level of reviewers have an influence.	119 undergraduate students 21 graduate students
	In total:	234 undergraduate students 134 graduate students

4.1 Experiment 1: Representation of Inconsistencies

In Daun et al. (2017) we reported an experiment to investigate whether reviews can benefit from the consolidated representation of inconsistencies within one Message Sequence Chart (MSC, cf. ITU (2016)) compared to the representation of inconsistencies in two distinct MSC. Therefore, the experiment compares two representations of inconsistencies: First, in one merged diagram. Second, in two distinct diagrams. The experiment was conducted with 68 graduate students. The experiment material was derived from an industrial sample specification of an avionics collision avoidance system.

The descriptive statistics of Experiment 1 are shown in Table 2. In this experiment only graduate students participated.

Regarding the question whether it is beneficial for manual reviews to first merge inconsistent behavioral properties into one integrated diagram compared to the review of inconsistent properties in separate diagrams, the experiment shows that such a model merging seems to have only limited impact on the effectiveness of the review. Regarding efficiency, when reviewing models with minor inconsistencies, merging inconsistent parts into one diagram can significantly improve the review's efficiency. Finally, the results show that regardless of the representation format (i.e., two separate diagrams or one merged diagram) the effectiveness of the review is considerably higher when reviewing diagrams with a high degree of consistency than when reviewing diagrams with a low degree of consistency.

4.2 Experiment 2: Representation of Recurring Instances

In Daun et al. (2020) we reported an experiment to investigate whether reviews of collaborative cyber-physical systems (CPS) specifications can benefit from using concrete instance-level MSC compared to the use of type-level MSC. The experiment was conducted with 55 undergraduate students and with 45 graduate students. The experiment material was also derived from the industrial sample specification of an avionics collision avoidance system.

The descriptive statistics of Experiment 2 are shown in Table 3. The experiment was conducted with undergraduate and graduate students.

The experiment shows that instance-level review diagrams as review artifacts are more expressive than the type-level diagrams. Expressiveness of instance-level review diagrams detailing more instances are also more expressive compared to instance-level review diagrams displaying fewer instances. Regarding effectiveness, the use of instance-level review diagrams showing three instances as review artifacts is more effective than the use of type-level diagrams. Efficiency and confidence were roughly equal for reviewing instance-level review diagrams and for reviewing type-level diagrams.

Table 2 Descriptive statistics of experiment 1

	Degree Program	N	Minimum	Maximum	Mean	Std. Deviation
Confidence Correct	Graduate	68	1.429	5.000	3.425	0.842
Confidence Incorrect	Graduate	68	1.000	4.800	3.019	0.846
Performance	Graduate	68	4.17%	91.67%	57.17%	18.84%
Self-Rated Experience	Graduate	67	1.000	4.000	2.377	0.725

Table 3 Descriptive statistics of experiment 2

	Degree Program	N	Minimum	Maximum	Mean	Std. Deviation
Confidence Correct	Undergraduate	55	1.250	5.000	4.096	0.736
	Graduate	45	2.222	5.000	4.370	0.620
Confidence Incorrect	Undergraduate	55	1.333	5.000	3.924	0.697
	Graduate	45	2.080	5.000	4.108	0.652
Performance	Undergraduate	55	19.44%	63.89%	42.67%	11.27%
	Graduate	45	16.67%	77.78%	52.04%	13.39%
Self-Rated Experience	Undergraduate	53	1.000	3.500	2.316	0.722
	Graduate	44	1.000	5.000	2.813	0.697

4.3 Experiment 3: Comparison Review Model vs Functional Design

In Daun et al. (2019a) we reported an experiment to investigate whether reviews of an automatically generated dedicated review model are beneficial compared to the review of the functional design of CPS. The experiment was conducted with 60 undergraduate students. The experiment material was derived from automotive case systems: door control unit and lane keeping support.

The descriptive statistics of Experiment 3 are shown in Table 4. In this experiment only undergraduate students participated.

Regarding the question whether the use of a generated review model can aid the review of the functional design, the controlled experiment provides evidence that confirms the hypothesis: The review of the review model is significantly more effective than the review of the original functional design. Participants were more self-confident regarding their decision and rated the use of the review model as more supportive. This means that the use of model transformations can be an effective means to improve the quality of reviews of model-based specifications. The experiment also showed that there seems to be no effect regarding the efficiency of the review. The review of both artifacts (i.e., the review model and the original functional design) seems equally efficient, so there is no impact of model transformations regarding the time reviewers need to make a correct decision.

4.4 Experiment 4: Comparison Review Model vs Original Specifications

In Daun et al. (2019b) we reported an experiment to investigate whether reviews of an automatically generated and consolidated dedicated review model are beneficial compared to the review of the behavioral requirements and the functional design of CPS. Furthermore, it investigates whether the experience and skill level of reviewers have an influence on

Table 4 Descriptive statistics of experiment 3

	Degree Program	N	Minimum	Maximum	Mean	Std. Deviation
Confidence Correct	Undergraduate	60	1.684	5.000	3.968	0.604
Confidence Incorrect	Undergraduate	60	1.455	4.444	3.463	0.454
Performance	Undergraduate	60	3.33%	86.67%	62.28%	15.78%
Self-Rated Experience	Undergraduate	57	1.000	4.000	2.471	0.738

the effectiveness and efficiency of the reviews. The experiment was conducted with 119 undergraduate students and with 21 graduate students.

The descriptive statistics of Experiment 4 are shown in Table 5. The experiment was conducted with undergraduate and graduate students.

The review of the review model is more effective, more efficient, more user confidence increasing and is subjectively perceived as more supportive than the review of behavioral requirements and functional design. This holds true for reviewers of different experience and skill levels. While graduate students are more effective than undergraduate students in reviewing the review model and the original specifications, the effect of the review model on effectiveness is the same for graduates and undergraduates.

5 Study Setup

This section introduces the study setup of the observational study. To ensure comparability with other empirical studies the structure of the section is based on the recommendations given in Wohlin et al. (2000) and Jedlitschka et al. (2008).

5.1 Goal

The objective of our study is to investigate whether students' self-rated experience and confidence are adequate means to predict students' performance.

Previous work has shown that self-rated experience and confidence are an adequate means to predict students' performance for programming tasks (cf. Section 2, e.g., Bunse (2006); Feigenspan et al. (2012); Bergersen et al. (2011); Layman et al. (2005)). Little work, however, has been done to study if this also holds for other areas of software engineering. To contribute to the body of research on the relation between self-rated experience, confidence, and performance, we studied the effects self-rated experience and confidence have on performance for model comprehension tasks. As it is important for validation and other activities in the context of model-based development to understand models, we want to investigate whether students' self-rated experiences and confidences are good predictors for their actual performance in model comprehension tasks.

Therefore, we defined two major research questions:

- **RQ1**: Is the self-rated confidence a student claims for answering model comprehension tasks a good predictor for correctness of the student's answer?

Table 5 Descriptive statistics of experiment 4

	Degree Program	N	Minimum	Maximum	Mean	Std. Deviation
Confidence Correct	Undergraduate	117	1.000	5.000	3.628	0.794
	Graduate	21	2.250	4.909	4.248	0.707
Confidence Incorrect	Undergraduate	119	1.000	4.889	3.510	0.761
	Graduate	21	1.800	5.000	3.918	1.017
Performance	Undergraduate	119	0.00%	87.50%	46.67%	15.98%
	Graduate	21	33.33%	83.33%	60.71%	15.4%
Self-Rated Experience	Undergraduate	117	1.000	4.333	2.499	0.733
	Graduate	21	1.500	4.667	3.071	0.802

- **RQ2**: Is self-rated experience a good predictor for students' performance in model comprehension tasks?

To do so, we analyze a set of comparable experiments we conducted with student participants (see Section 4). As the experiments were conducted with graduate and undergraduate students, this also allows investigating the relations between students' performance and students' perception of their performance separately for graduate and undergraduate students (which can also be seen as a characteristic of experience, see Section 3.1). Hence, we also want to determine for RQ1 and RQ2 if there is a difference between undergraduate and graduate students with respect to their ability to assess the correctness of their answers and their performance. Leading to:

- **RQ3**: Does graduate students' self-rated confidence better reflect the correctness of their answers compared to undergraduate students?
- **RQ4**: Does the self-rated experience of graduate students better predict their performance compared to undergraduate students?

5.2 Variables

Table 6 introduces the response variables and the explanatory variables of the observational study. Note that these are used in the individual experiments as dependent and independent variables or are calculated from the dependent and independent variables of the individual experiments. In the following each variable is explained in more detail.

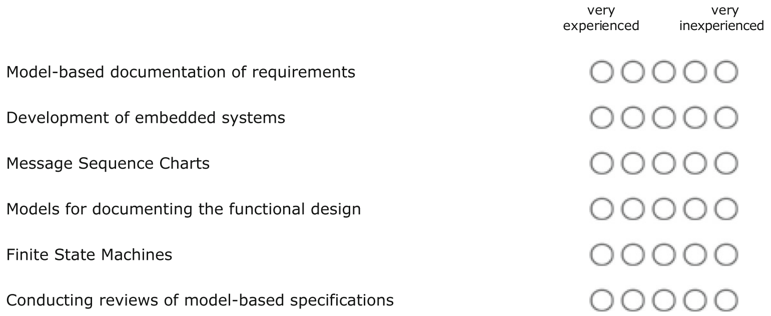
Response variables are **Correctness** and **Performance** as we want to investigate whether the likelihood of performing better (or on an item level: making a correct decision) can be explained with the explanatory variables: **Confidence**, **Self-Rated Experience**, and **Degree Program**.

Correctness is measured on an ordinal scale, an answer given by a participant is either *Correct* or *Incorrect*. We define **Performance** as the ratio of correct answers and all answers given. **Performance** is measured on a ratio scale from 0% to 100%.

Regarding the confidences, for each answer the participant is asked to self-rate their confidence in the correctness of the decision they made. As shown in Figs. 2 and 3 confidence

Table 6 Observed variables

Name	Type	Scale	Characteristic	Description
Correctness	Response	Ordinal	Correct incorrect	Students' answers are either correct or incorrect
Performance	Response	Ratio	0%-100%	The ratio of correctly answered questions
Confidence	Explanatory	Ratio	1-5	The average confidence a participant claims for giving answers
Self-Rated Experience	Explanatory	Ratio	1-5	The average result of participants' self-rated experiences
Degree Program	Explanatory	Nominal	Undergraduate graduate	Participants are either enrolled in an undergraduate or a graduate degree program



	very experienced				very inexperienced
Model-based documentation of requirements	○	○	○	○	○
Development of embedded systems	○	○	○	○	○
Message Sequence Charts	○	○	○	○	○
Models for documenting the functional design	○	○	○	○	○
Finite State Machines	○	○	○	○	○
Conducting reviews of model-based specifications	○	○	○	○	○

Fig. 1 Example of items measuring the experience for experiment 3

is measured on a 5-point semantic differential scale. For **Confidence Correct** we calculate the mean of the measured confidence values for all answers given that were *Correct*. For **Confidence Incorrect** we calculate the mean of the measured confidence values for all answers given that were *Incorrect*. **Confidence Correct** and **Confidence Incorrect** are measured on a ratio scale with 1 as the minimum and 5 as the maximum value.

To measure **Self-Rated Experience**, the participants are asked questions to self-rate their experience subsequent to experiment execution. For instance, participants are asked for their experience with the involved modeling languages or the kind of tasks they had to conduct. Figure 1 shows an example for the questions asked for Experiment 3. Each item is measured on a 5-point semantic differential scale, where 1 means *very inexperienced* and 5 means *very experienced*. **Self-Rated Experience** is then calculated by averaging the value of all these semantic differential scale items. Hence, **Self-Rated Experience** itself is measured on a ratio scale with 1 as the minimum and 5 as the maximum.

The degree program has two levels, participants are either enrolled in an *Undergraduate* degree program or in a *Graduate* degree program. In our case graduate students always refer to the Master level, not to Ph.D. students.

Please note, we measure experience and confidence on a 5-point semantic differential scale.[3] Table 6 describes these variables to be defined on a ratio scale. This is because we aggregated the data by calculating the average of multiple items. For confidence, we calculate the average confidence of multiple tasks. For self-rated experience, the self-rated experience construct is determined by the average value of multiple questionnaire items related to different areas of expertise. While this is a very common approach in many reported experiments (e.g., El Emam and Madhavji (1996); Sinha and Smidts (2006); Polančič et al. (2010)), some object to this practice as they argue that the differences between different points on the semantic differential scale cannot be assumed to be equal (cf. Jamieson (2004)). Therefore, in the appendix of this paper you will find a non-parametric evaluation of the hypotheses that treats confidence and self-rated experience as ordinal values. As can be seen, the treatment of these variables as ordinal values as opposed to ratio values has in this case no effect on the acceptance of hypotheses.

[3] In the original experiment reports we erroneously reported to have measured experience and confidence on Likert-scales (cf. Likert (1932)). However, we actually used a semantic differential scale (cf. Osgood et al. (1957)) as we did not measure agreement with a statement as is required for Likert-scales. As this is a common misunderstanding of Likert-scales, we also corrected claims in the related work section where authors reported the use of Likert-scales but actually used semantic differential scales.

5.3 Hypothesis Formulation

5.3.1 Hypotheses Regarding Underlying Assumptions

In theory, we assume that more advanced students are more confident, rate their experience higher, and perform better than inexperienced students. One objectively measurable difference in the experience of students are the degrees they already received. In our case, this is the distinction between graduate students and undergraduate students. Therefore, we define following hypotheses:

HA.1_A: *Graduate students perform better than undergraduates.*

I.e., $Perform_{Grad} > Perform_{Ugrad}$

HA.2_A: *Graduate students rate their confidence higher than undergraduates.*

I.e., $Conf_{Grad} > Conf_{Ugrad}$

HA.3_A: *Graduate students rate their experience higher than undergraduates.*

I.e., $Exp_{Grad} > Exp_{Ugrad}$

The corresponding null hypotheses are:

HA.1_0: *There is no difference in performance between graduate and undergraduate students.*

I.e., $Perform_{Grad} = Perform_{Ugrad}$

HA.2$_0$**:** *There is no difference in confidence between undergraduate and graduate students.*

I.e., $Conf_{Grad} = Conf_{Ugrad}$

HA.3_0: *There is no difference in self-rated experience between graduate and undergraduate students.*

I.e., $Exp_{Grad} = Exp_{Ugrad}$

5.3.2 Hypotheses Regarding Confidence and Correctness (RQ1)

Regarding **RQ1**, we want to investigate whether the students' confidence in decision making is a good predictor for the correctness of the result. Therefore, we assume that students are more confident when giving a correct answer. Hence, the alternative hypothesis is:

H1.1_A: *Students rate their confidence higher when giving a correct answer than when giving an incorrect answer.*

I.e., $ConfCorrect > ConfIncorrect$

The corresponding null hypothesis is:

H1.1_0: *There is no difference in confidence between correct and incorrect answers.*

I.e., $ConfCorrect = ConfIncorrect$

5.3.3 Hypotheses Regarding Self-Rated Experience and Performance (RQ2)

Regarding **RQ2**, we want to investigate whether better performing students rate their experience higher. Assuming that students can assess whether they have given a correct or an incorrect answer, it follows that students whose performance was better should self-rate their experience higher than worse performing students. Therefore, we define:

H2.1_A: *Students' performance is positively correlated with their self-rated experience.*

The corresponding null hypothesis is:

H2.1_0: *There is no correlation between students' performance and their self-rated experience.*

5.3.4 Hypotheses Regarding the Impact of the Degree Program on Confidence and Correctness (RQ3)

For **RQ3**, we want to broaden our investigation of **RQ1** by taking the experience level of the degree program into account. We expect graduate students to be better than undergraduate students at assessing whether they have given a correct answer or an incorrect one. In detail, this means that we expect graduate students to have a higher average confidence for correct answers compared to undergraduate students; and we expect graduate students to have a lower average confidence for incorrect answers than undergraduate students, indicating that graduate students are more aware of the correctness of their answers and rate their confidence accordingly. Thus, we define:

H3.1_A: *Graduate students rate their confidence higher when giving a correct answer than undergraduates do when giving a correct answer.*
I.e., $ConfCorrect_{Grad} > ConfCorrect_{Ugrad}$

H3.2_A: *Graduate students rate their confidence lower when giving a incorrect answer than undergraduates do when giving an incorrect answer.*
I.e., $ConfIncorrect_{Grad} < ConfIncorrect_{Ugrad}$

The corresponding null hypotheses are:

H3.1_0: *There is no difference in Confidence Correct between undergraduate and graduate students.*
I.e., $ConfCorrect_{Grad} = ConfCorrect_{Ugrad}$

H3.2_0: *There is no difference in Confidence Incorrect between undergraduate and graduate students.*
I.e., $ConfIncorrect_{Grad} = ConfIncorrect_{Ugrad}$

5.3.5 Hypotheses Regarding the Impact of the Degree Program on Self-Rated Experience and Performance (RQ4)

For **RQ4**, we also want to broaden our investigation of **RQ2** by taking the experience level of the degree program into account. Therefore, we can assume that the more advanced graduate students are better at estimating their experience-level based on their actual performance. Hence, we define:

H4.1_A: *Performance and self-rated experience are more strongly correlated for graduates compared to undergraduates.*

The corresponding null hypothesis is:

H4.1_0: *There is no difference in the correlations between students' performance and self-rated experience between graduates and undergraduates.*

5.3.6 Summary of Hypotheses

The aforementioned null hypotheses correspond to two directed alternative hypotheses. However, so far, we only reported the directed alternative hypotheses Hx_A, which are in accordance with our expectations. Table 7 summarizes all hypotheses and complements the alternative hypotheses for the opposite direction, which we refer to as Hx_B.

5.4 Experiment Designs

As mentioned above, we use data collected from several experiments. The experiments were originally conducted to compare different types of model-based specifications with respect to their suitability for manual reviews. Participants were shown excerpts from specifications and asked whether certain natural language stakeholder intentions were reflected in that particular model or not. For each stakeholder intention, participants were also asked to rate on a 5-point semantic differential scale how confident they are that the answer they have given is correct. After answering all review questions, participants were asked about their experiences with model-based engineering in general and with the notations used in the experiment.

To answer **RQ1** we compare each student's average confidence for giving correct answers with their average confidence for giving incorrect answers. Therefore, correctness is a within subject factor. Considering **RQ3** we also distinguish between graduate and undergraduate students. Participants varied between the two groups. Therefore, degree program is a between-subject factor.

To answer **RQ2**, we check whether the students' self-rated experience is correlated to their performance. Considering **RQ4** we again distinguish between graduate and undergraduate students. As we did not conduct a longitudinal study, participants varied between the two groups. Therefore, self-rated experience and performance are between subject factors.

5.5 Participants

The experiments were conducted with graduate and undergraduate students, who were recruited within university courses on requirements engineering. The university offers two courses on requirements engineering: one undergraduate course that is compulsory for systems engineering majors and optional for information systems majors and focuses on documentation of requirements, and one graduate course that is optional for all participating students and focuses on requirements analysis. The undergraduate course is not a prerequisite for the graduate course. The two courses are offered in different terms, hence no student can be enrolled in both at the same time. All experiments were first conducted in the

Table 7 Hypotheses

	Hypotheses
HA.1_0	*There is no difference in performance between graduate and undergraduate students.*
HA.1_A	*Graduate students perform better than undergraduates.*
HA.1_B	*Undergraduate students perform better than graduates.*
HA.2_0	*There is no difference in confidence between undergraduate and graduate students.*
HA.2_A	*Graduate students rate their confidence higher than undergraduates.*
HA.2_B	*Undergraduate students rate their confidence higher than graduates.*
HA.3_0	*There is no difference in self-rated experience between graduate and undergraduate students.*
HA.3_A	*Graduate students rate their experience higher than undergraduates.*
HA.3_B	*Undergraduate students rate their experience higher than graduates.*
H1.1_0	*There is no difference in confidence between correct and incorrect answers.*
H1.1_A	*Students rate their confidence higher when giving a correct answer than when giving an incorrect answer.*
H1.1_B	*Students rate their confidence higher when giving an incorrect answer than when giving a correct answer.*
H2.1_0	*There is no correlation between students' performance and their self-rated experience.*
H2.1_A	*Students' performance is positively correlated with their self-rated experience.*
H2.1_B	*Students' performance is negatively correlated with their self-rated experience.*
H3.1_0	*There is no difference in Confidence Correct between undergraduate and graduate students.*
H3.1_A	*Graduate students rate their confidence higher when giving a correct answer than undergraduates do when giving a correct answer.*
H3.1_B	*Undergraduate students rate their confidence higher when giving a correct answer than graduates do when giving a correct answer.*
H3.2_0	*There is no difference in Confidence Incorrect between undergraduate and graduate students.*
H3.2_A	*Graduate students rate their confidence lower when giving a incorrect answer than undergraduates do when giving an incorrect answer.*
H3.2_B	*Undergraduate students rate their confidence lower when giving a incorrect answer than graduates do when giving an incorrect answer.*
H4.1_0	*There is no difference in the correlations between students' performance and their self-rated experience between graduates and undergraduates.*
H4.1_A	*Performance and self-rated experience are stronger correlated for graduates compared to undergraduates.*
H4.1_B	*Performance and self-rated experience are stronger correlated for undergraduates compared to graduates.*

graduate course, thus ensuring no students participated in the same experiment twice. However, some students first participated in one experiment in the undergraduate course and then in a different experiment in the graduate course.

5.5.1 Graduate Students

Graduate participants were recruited from master-level university courses on requirements engineering. The participants were mainly holding bachelor degrees in 'Systems Engineering' or ’Information Systems’ and were enrolled in the master-level degree programs of ’Systems Engineering’ or ’Information Systems’.

The course’s syllabus consists of goals, scenarios, essential system analysis, requirements validation and management. Model-based requirements engineering techniques are an integral part of the course; among others, students are taught scenario definition using MSC. Hence, at the point where the experiment takes place the students have knowledge of the modeling languages used as well as of conducting reviews as a requirements validation activity. We did not conduct additional briefings to avoid threats from hypothesis guessing and the like. However, debriefings were conducted to relate the experiments to the course and deepen an understanding of requirements validation activities.

5.5.2 Undergraduate Students

Undergraduate participants were recruited from bachelor-level requirements engineering courses. The participants were mainly enrolled in bachelor-level degree programs for ’Systems Engineering’, ’Information Systems’, or ’Business Studies’.

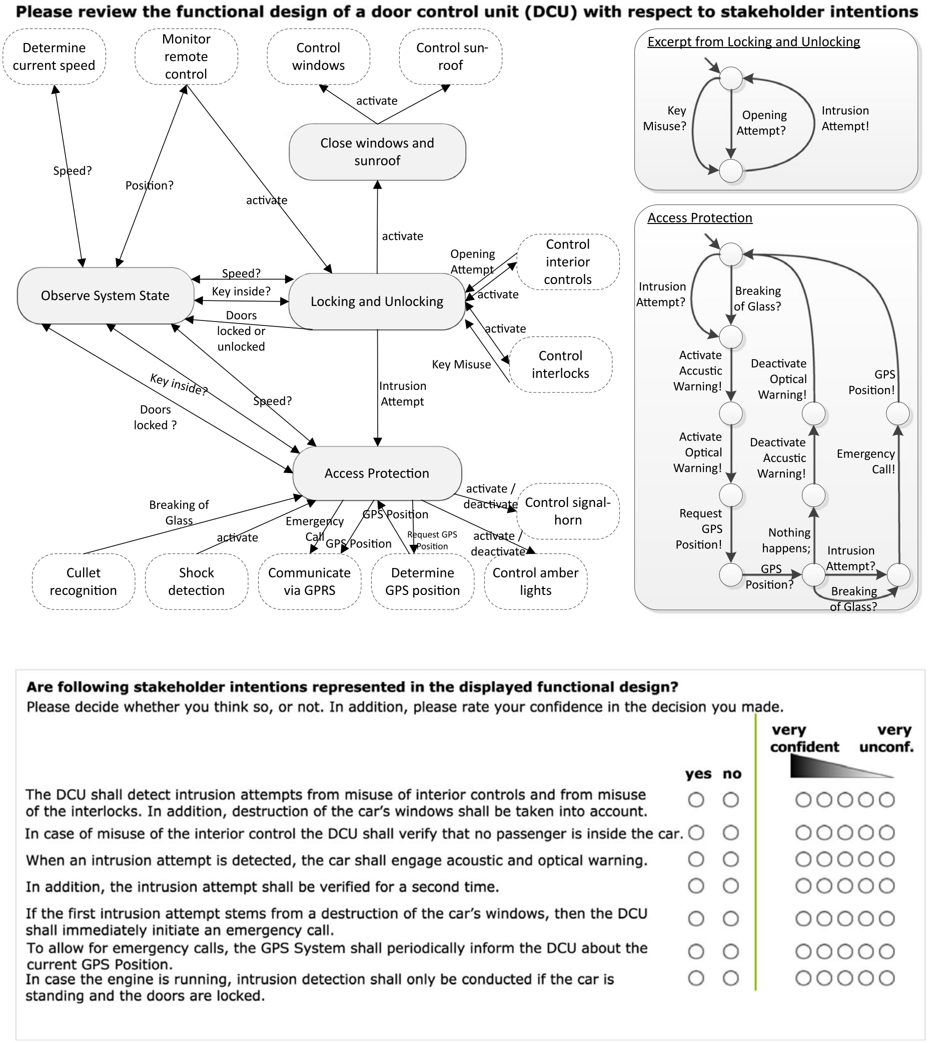
Much akin to the graduate course, model-based requirements engineering is an important part of the undergraduate course. Furthermore, requirements validation is also part of the course, however to a lesser extent than in the graduate course. Therefore, undergraduate students also have a basic understanding of the modeling languages used and of requirements validation tasks. Also in this case, we refrained from upfront briefings but integrated debriefings in the course syllabus.

5.6 Experiment Materials

The individual experiments’ experimental setups consist of several reviewing tasks and a post-hoc questionnaire. Each participant reviewed specification excerpts and had to decide whether certain statements are correctly displayed within the specification. For the specification excerpts, we used various industrial sample specifications as sources for the experiment material. The participants had to conduct reviews of excerpts from model-based specifications of the industrial case systems. Additionally, a set of natural language statements, which had to be evaluated by the participants, was given. The participants had to decide whether these statements were properly described by the specifications. In addition to each statement, the participants were asked to rate their confidence in decision making on a 5-point semantic differential scale.

Figures 2 and 3 show excerpts from Experiment 3. Shown are diagrammatic representations (i.e., in case of Fig. 2 an excerpt from the functional design of an automotive door control unit and in case of Fig. 3 an excerpt from a review model for an automotive lane keeping support) and natural language statements. The participants were asked to investigate whether these natural language statements are correctly incorporated in the models shown. As these statements were either true, i.e. the aspect was displayed in the model, or not true, i.e. the aspect was not displayed in the model, an answer was judged correct if the participant either identified a true statement as true or an untrue statement as not true.

After the reviewing tasks, each participant was asked to rate their level of experience in several categories related to the reviewing tasks. Self-rated experience is determined as the mean of the measurements of several items rated on a 5-point semantic differential scale (see Fig. 1 for an example from Experiment 3). The self-rated experience was explicitly


Please review the functional design of a door control unit (DCU) with respect to stakeholder intentions

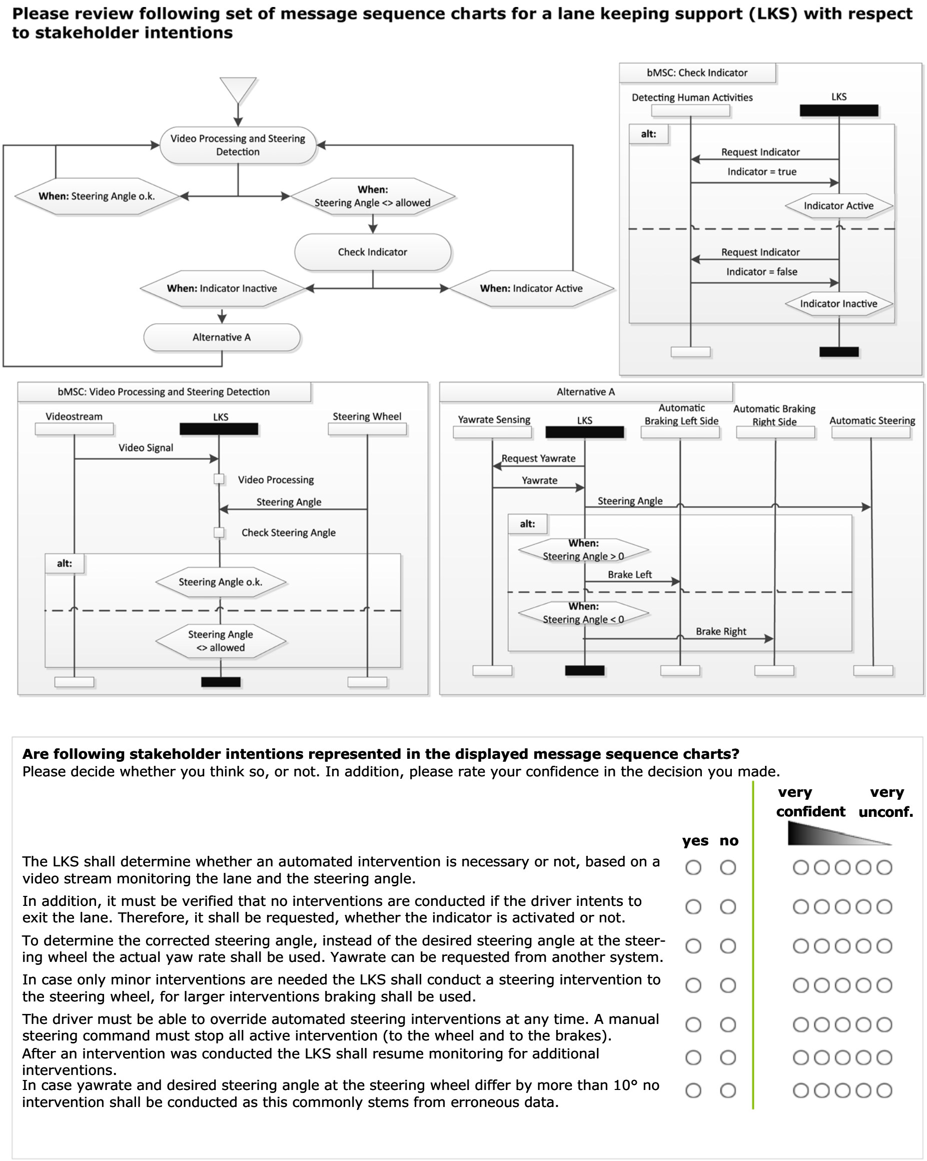
Are following stakeholder intentions represented in the displayed functional design?
Please decide whether you think so, or not. In addition, please rate your confidence in the decision you made.

	yes	no	very confident … very unconf.
The DCU shall detect intrusion attempts from misuse of interior controls and from misuse of the interlocks. In addition, destruction of the car's windows shall be taken into account.	○	○	○○○○○
In case of misuse of the interior control the DCU shall verify that no passenger is inside the car.	○	○	○○○○○
When an intrusion attempt is detected, the car shall engage acoustic and optical warning.	○	○	○○○○○
In addition, the intrusion attempt shall be verified for a second time.	○	○	○○○○○
If the first intrusion attempt stems from a destruction of the car's windows, then the DCU shall immediately initiate an emergency call.	○	○	○○○○○
To allow for emergency calls, the GPS System shall periodically inform the DCU about the current GPS Position.	○	○	○○○○○
In case the engine is running, intrusion detection shall only be conducted if the car is standing and the doors are locked.	○	○	○○○○○

Fig. 2 Example experiment materials - functional design of an automotive door control unit

measured after the tasks have been conducted by participants, to ensure that participants can consider their perceived performance in rating their experience. This rating was also conducted on a 5-point semantic differential scale.

The experiments were conducted using online questionnaires. The experiments were designed to last about 30 minutes. This was done to minimize the threat of participants' dropping out of the experiment because of lost interest. Students typically participated from home, as a) we wanted to reduce interaction effects between participants due to limited space available in the class room and b) sufficient equipment was not available to allow for in-class participation.



Please review following set of message sequence charts for a lane keeping support (LKS) with respect to stakeholder intentions

Are following stakeholder intentions represented in the displayed message sequence charts?
Please decide whether you think so, or not. In addition, please rate your confidence in the decision you made.

	yes	no	very confident … very unconf.
The LKS shall determine whether an automated intervention is necessary or not, based on a video stream monitoring the lane and the steering angle.	○	○	○○○○○
In addition, it must be verified that no interventions are conducted if the driver intents to exit the lane. Therefore, it shall be requested, whether the indicator is activated or not.	○	○	○○○○○
To determine the corrected steering angle, instead of the desired steering angle at the steering wheel the actual yaw rate shall be used. Yawrate can be requested from another system.	○	○	○○○○○
In case only minor interventions are needed the LKS shall conduct a steering intervention to the steering wheel, for larger interventions braking shall be used.	○	○	○○○○○
The driver must be able to override automated steering interventions at any time. A manual steering command must stop all active intervention (to the wheel and to the brakes).	○	○	○○○○○
After an intervention was conducted the LKS shall resume monitoring for additional interventions.	○	○	○○○○○
In case yawrate and desired steering angle at the steering wheel differ by more than 10° no intervention shall be conducted as this commonly stems from erroneous data.	○	○	○○○○○

Fig. 3 Example experiment materials - review model for an automotive lane keeping support

5.7 Analysis Methods

We analyzed the data by calculating common descriptive statistic parameters. The measured construct of self-rated experience is calculated as the mean of multiple items that are related to experience in different fields of expertise. Therefore, it must be evaluated whether these single items do not contradict each other and it is fair to claim that these can be aggregated. To ensure item reliability of self-rated experience, we computed Cronbach's alpha.

To compare graduate students' performance to undergraduate students' performance, graduate students' confidence to undergraduate students' confidence, and graduate students' self-rated experience to undergraduate students self-rated experience, we conducted independent t tests. To compare students' confidence between correct and incorrect answers, we conducted a paired-samples t test. For examining the relationship between students' performance and their self-rated experiences, we evaluated partial correlations. This allows us to examine the relationship between students' performance and their self-rated experience while controlling for degree program. We compare graduate students' confidence when giving a correct answer to undergraduate students' confidence when giving a correct answer and graduate students' confidence when giving an incorrect answer to undergraduate students' confidence when giving an incorrect answer using independent t tests. We also evaluate correlations between performance and self-rated experience for graduate and undergraduate students separately and compared these correlations using Fisher's z. Fisher's z is used to test if two correlations differ significantly (cf. Diedenhofen and Musch (2015)).

6 Results

This section reports the results of the observational study. The descriptive statistics are presented in Section 6.1. Subsequently, Section 6.2 briefly summarizes the data set preparation before Section 6.3 details the results of the conducted hypothesis tests. Note that we report the results of the observational study across all experiments at once. For detailed descriptive statistics of each individual experiment, please refer to Section 4.

6.1 Descriptive Statistics

Table 8 shows the descriptive statistics. There were two undergraduate students who gave no correct answer. Consequently, their average confidence for giving correct answers could not be determined. Similarly, there were nine students (seven undergraduates and two graduate students) who chose not to disclose their experiences.

Table 9 shows the results of Pearson correlations between the different variables. As can be seen the various confidence variables are unsurprisingly strongly correlated. Self-rated experience is weakly correlated with confidence. Performance is weakly correlated with confidence correct and negligibly correlated with confidence incorrect.

Table 8 Descriptive statistics

		N			
	Degree Program	Valid	Missing	Mean	Std. Deviation
Confidence Correct	Undergraduate	232	2	3.827	0.761
	Graduate	134	0	3.872	0.876
Confidence Incorrect	Undergraduate	234	0	3.596	0.701
	Graduate	134	0	3.525	0.963
Performance	Undergraduate	234	0	49.74%	16.70%
	Graduate	134	0	56.00%	16.83%
Self-Rated Experience	Undergraduate	227	7	2.449	0.732
	Graduate	132	2	2.633	0.773

Table 9 Pearson correlations

		Confidence Overall	Confidence Correct	Confidence Incorrect	Performance	Self-Rated Experience
Confidence overall	Pearson Correlation	1	.771**	.848**	.062	-.264**
	Sig. (2-tailed)		.000	.000	.235	.000
	N	368	366	368	368	359
Confidence correct	Pearson Correlation	.771**	1	.826**	.289**	-.271**
	Sig. (2-tailed)	.000		.000	.000	.000
	N	366	366	366	366	357
Confidence incorrect	Pearson Correlation	.848**	.826**	1	.142**	-.259**
	Sig. (2-tailed)	.000	.000		.006	.000
	N	368	366	368	368	359
Performance	Pearson Correlation	.062	.289**	.142**	1	-.051
	Sig. (2-tailed)	.235	.000	.006		.336
	N	368	366	368	368	359
Self-Rated experience	Pearson Correlation	-.264**	-.271**	-.259**	-.051	1
	Sig. (2-tailed)	.000	.000	.000	.336	
	N	359	357	359	359	359

**. Correlation is significant at the .01 level (2-tailed).

6.1.1 Confidence

As can also be seen from the boxplots in Fig. 4, in mean students express a higher confidence when giving a correct answer than when giving an incorrect answer.

When giving a correct answer graduate students express a higher confidence (M = 3.872, SD = .876) than undergraduate students (M = 3.827, SD = .761). When giving an incorrect answer, however, graduate students (M = 3.525, SD = .963) express a lower confidence than undergraduate students (M = 3.596, SD = .701).

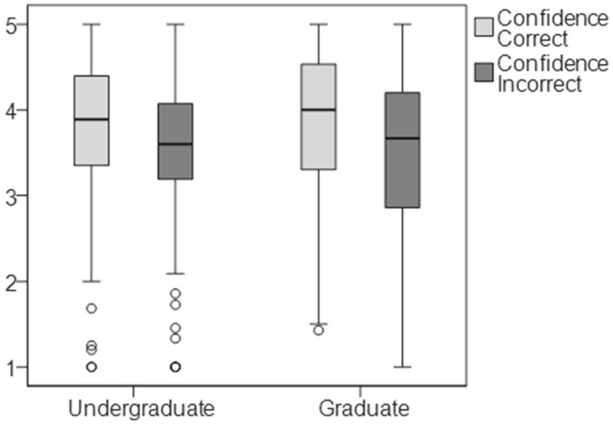


Fig. 4 Confidence for correct and incorrect answers depending on degree program

6.1.2 Performance

It can be seen from the boxplots in Fig. 5, that in mean graduate students perform better (M = 56.001%, SD = 16.832%) than undergraduate students (M = 49.735%, SD = 16.699%).

6.1.3 Experience

As can be seen from the boxplots in Fig. 6, undergraduate students rate their experience (M = 2.449, SD = .732) lower than graduate students (M = 2.633, SD = .773).

6.1.4 Comparisons

The scatterplot in Fig. 7 illustrates average confidence for correct and incorrect answers for each individual student. Remember that confidence was measured on a 5-point semantic differential scale ranging from one to five, with one representing highly unconfident and five representing highly confident. It can be seen that most participants had a higher confidence when giving a correct answer than when giving an incorrect answer. However, a few participants had a wide gap between their confidence averages.

The scatterplot in Fig. 8 illustrates the individual students' performances and their self-rated experiences. It can be seen that there seems to be no strong connection between a student's self-rated experience and their performance.

6.2 Data Set Preparation

We combined the data sets from four previously conducted and reported experiments. For the original analyses, we removed outliers that hinted at irregularities such as participants completing the experiment too quickly, which raised suspicion that these participants answered questions randomly to finish as fast as they could. This was the case for five participants of Experiment 1, who took less than five minutes to review 34 natural language stakeholder intentions, and six participants of Experiment 4, who took less than one minute to review twelve natural language stakeholder intentions. We removed all data from these participants. Beyond outlier removal in the original experiments, we removed no further data sets for the observational study.

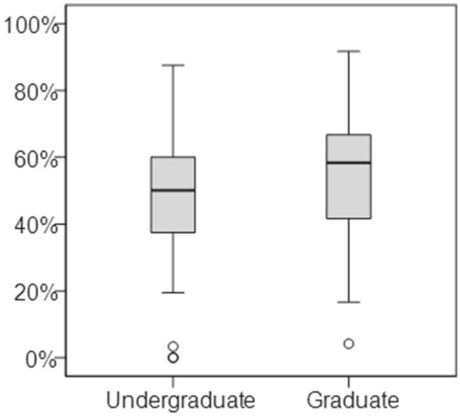


Fig. 5 Performance of undergraduate and graduate students

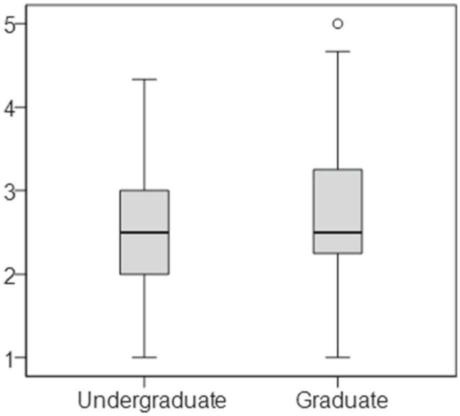


Fig. 6 Experience of undergraduate and graduate students

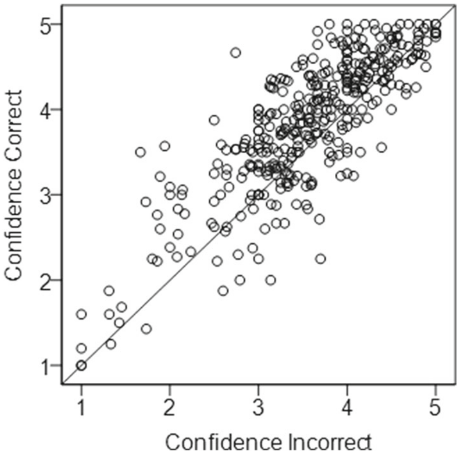


Fig. 7 Scatterplot comparing each student's average confidence in correct and incorrect answers

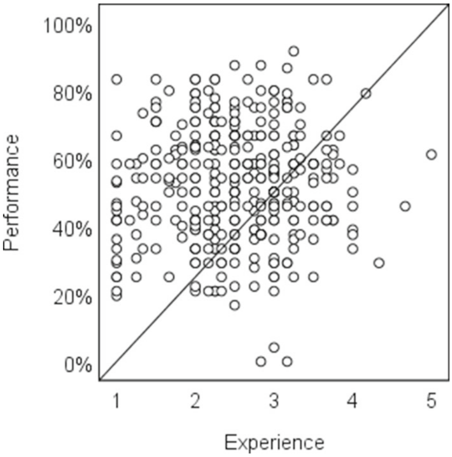


Fig. 8 Scatterplot comparing each student's performance and self-rated experience

6.3 Hypothesis Testing

The data collected is nearly normally distributed. Table 10 shows the results of the Kolmogorov-Smirnov and the Shapiro-Wilks tests. As can be seen, for some variables the tests indicate non-normality, however, in large samples these tests are known to show significant results even for small and unimportant effects, i.e. small deviations from a normal distribution (Field (2013)). Additionally, according to the central limit theorem, for samples larger than 30, the sampling distribution of the data will take the shape of a normal distribution regardless of the shape of the population from which the sample was drawn, consequently the research literature recommends using parametric tests for analyzing such data (cf. Lumley et al. (2002)). Due to this, we are confident in the reliability of parametric tests in this setting (cf. Field (2013)). In addition, we had a need to investigate the reliability of our measurement for self-rated experience. Self-rated experience is determined as the mean of the measurements of several questions rated on a 5-point semantic differential scale. Hence, it must be ensured that the aggregation of these different questions to one measurement for experience is fair (i.e., that the different items are not inconsistent. We, therefore, conducted tests for Cronbach's alpha. We determined a good reliability of $\alpha(6) = .841$. Consequently, we assume our questionnaire for determining self-rated experience to be sufficiently reliable (cf. DeVellis (2017)). Note that this only means that the various items we use to define the construct self-rated experience are not contradictory. However, it does not prove that they actually measure participants experience nor that they are indeed causally related, as we will discuss in Section 7.2.

6.3.1 Hypothesis Tests Regarding Underlying Assumptions

Regarding the assumption that graduate students perform better than undergraduates, our results show that on average, graduate students (M = 56.00%, SD = 16.83%) performed highly significantly better than undergraduates (M = 49.74%, SD = 16.70%), t(366) = 3.45, p = .001. Therefore, we **accept HA**.1_A.

Table 10 Tests of normality

		Kolmogorov-Smirnov^a			Shapiro-Wilk		
		Statistic	df	Sig.	Statistic	df	Sig.
Confidence	Undergraduate	.065	234	.019	.972	234	.000
Overall	Graduate	.095	134	.005	.961	134	.001
Confidence	Undergraduate	.062	232	.030	.944	232	.000
Correct	Graduate	.118	134	.000	.924	134	.000
Confidence	Undergraduate	.074	234	.003	.950	234	.000
Incorrect	Graduate	.092	134	.007	.963	134	.001
Performance	Undergraduate	.043	234	.200*	.989	234	.065
	Graduate	.092	134	.007	.981	134	.061
Self-Rated	Undergraduate	.087	227	.000	.977	227	.001
Experience	Graduate	.114	132	.000	.977	132	.026

* This is a lower bound of the true significance.

a Lilliefors Significance Correction

Regarding the assumption that graduate students have higher confidence than undergraduates, our results show that on average, graduate students (M = 3.54, SD = 0.88) are not significantly more confident than undergraduates (M = 3.46, SD = 0.79), t(254.50) = 0.83, p = .409. Therefore, we **cannot reject** **HA**.2_0.

Regarding the assumption that graduate students self-rate their experience higher than undergraduates, we found out that on average, undergraduate students (M = 2.45, SD = .73) rated their experience significantly lower than graduate students (M = 2.63, SD = .77), t(357) = 2.24, p = .025. Hence, we can **accept** **HA**.3_A.

6.3.2 Hypothesis Tests Regarding Confidence and Correctness (RQ1)

On average, confidence was higher when giving a correct answer (M = 3.84, SD = 0.80) than when giving an incorrect answer (M = 3.57, SD = 0.81). This difference was highly significant t(365) = 11.05, $p < .001$. Hence, we **accept** **H1**.1_A.

6.3.3 Hypothesis Tests Regarding Self-Rated Experience and Performance (RQ2)

A partial correlation was conducted to evaluate the relationship between performance and self-rated experience while controlling for degree program. When we control degree program on the relationship between performance and self-rated experience, we find the following partial correlation r = .03, p (one-tailed) = .282. This shows that independent of the degree program, better performing students do not rate their experience higher than weaker performing students. We therefore **cannot reject** **H2**.1_0.

6.3.4 Hypothesis Tests Regarding the Impact of the Degree Program on Confidence and Correctness (RQ3)

On average, confidence when giving a correct answer was higher for graduate students (M = 3.87, SD = 0.88) than for undergraduate student (M = 3.83, SD = 0.76). This difference was not significant t(364) = 0.512, p = .609. Therefore, we **cannot reject** **H3**.1_0.

On average, confidence when giving an incorrect answer was lower for graduate students (M = 3.53, SD = 0.96) than for undergraduate students (M = 3.60, SD = 0.70). This difference was not significant t(214,75) = -0.738, p = .461. Thus, we **cannot reject** **H3**.2_0.

6.3.5 Hypothesis Tests Regarding the Impact of the Degree Program on Self-Rated Experience and Performance (RQ4)

For undergraduates, performance was not significantly correlated with self-rated experience r = .09, p (one-tailed) = .099. Similarly, for graduate students, performance was not significantly correlated with self-rated experience r = -.06, p (one-tailed) = .250. A comparison of these two correlations using Fisher's z shows no significant difference between the correlations z = 1.31, p = .189. Therefore, we **cannot reject** **H4**.1_0.

6.3.6 Summary of Hypothesis Tests

Table 11 summarizes the results of the hypothesis tests. The accepted alternative hypotheses as well as the not rejected null hypotheses are recapped. Section 6 will subsequently elaborate on the results and discuss the implications and inferences of these.

Table 11 Overview of accepted alternative hypotheses and not rejected null hypotheses

Hypothesis	Result	Finding
$HA.1_A$	Accepted	On average graduate students performed better than undergraduates
$HA.2_0$	Not rejected	Confidence does not vary significantly between graduates and undergraduates
$HA.3_A$	Accepted	On average graduate students rated their experience higher than undergraduates
$H1.1_A$	Accepted	Student confidence is higher for correct answers than for incorrect answers
$H2.1_0$	Not rejected	There is no correlation between students' performance and their self-rated experience
$H3.1_0$	Not rejected	There is no difference in confidence correct between undergraduate and graduate students
$H3.2_0$	Not rejected	There is no difference in confidence incorrect between undergraduate and graduate students
$H4.1_0$	Not rejected	The correlations between performance and self-rated experience do not vary between graduate and undergraduate students

7 Discussion

7.1 Implications

While we discuss the implications of the hypothesis tests in this section, we must keep in mind that the experiments dealt with tasks related to perceiving, interpreting, and decision making on conceptual models. Hence, we will limit the interpretation of the results to students' self-perception regarding their experience and their confidence in contrast to their performance to the field of conceptual modeling.

Major findings include:

- In **RQ1** we aimed at finding out whether the confidence students claim for answering a question is a good predictor for the correctness of this question. Therefore, we determined whether students are aware if an answer they have given is likely to be correct or not. The results show that for concrete tasks students can indeed assess whether their answer is more likely to be correct or incorrect. Hence, self-rated confidence for a concrete task can be seen as good predictor for the correctness of this task.
- Regarding **RQ2**, we wanted to determine whether self-rated experience is a good predictor for performance. Therefore, we investigated if better performing students rate their experience higher. First, we can state that there is no significant correlation between students' performance and their self-rated experience. Consequently, we can state self-rated experience is not a good predictor for students' performance.
- For **RQ3**, we investigated whether there is a difference in the relation between confidence and correctness for graduate and undergraduate students. Graduate students and undergraduate students are comparably confident independent of the correctness of their answers. We did neither find graduate students to be more confident when giving a correct answer compared to undergraduate students nor did we find graduate students to be less confident when giving an incorrect answer compared to undergraduate students. Hence, there is no effect discernable with respect to the degree program.
- For **RQ4**, we investigated whether there is a difference in the relation of self-rated experience and performance between graduate and undergraduate students. Graduate

students, which must be considered more experienced than undergraduate students, rate their experience higher than undergraduate students. Hence, students seem to be somewhat aware of their overall experience. This finding is in line with results for the assumption that graduate students in fact perform better than undergraduate students. However, the degree has an higher influence on self-rated experience than the actual performance of students.

Although there is a significant difference between self-rated experience of graduates and undergraduates, this does not lead graduate students to more accurately rate their experience with respect to their performance than undergraduates. Hence, graduate students (who are more experienced than undergraduates) are better performing and rate their experience higher than undergraduate students. However, their self-rated experience is not better correlated with their actual performance.

In summary, for self-rated experience, we can, thus, conclude that self-rated experience is not a good predictor for performance in model comprehension tasks. This results from the finding that no significant difference between self-rated experience and performance was found. Furthermore, this is substantiated by the fact that - although graduate students do rate their experience higher - the correlations between self-rated experience and performance do not vary between graduate and undergraduate students.

However, the fact that graduate students have a better performance and rate their experience higher indeed shows that students advance due to their study and, hence, graduate students can be seen as more experienced and that they are aware that they are more experienced. Nevertheless, there is no strong correlation between self-rated experience and performance on a within-subject basis such that self-rated experience is not a good predictor.

When it comes to confidence, we found that confidence can be used as predictor for correctness of an answer in model comprehension tasks. Regarding the difference between graduates and undergraduates, graduates are neither significantly more confident in giving correct answers nor significantly less confident in giving incorrect answers. However, for both groups confidence in a task is related to the correctness of the task. Thus, self-rated confidence can be seen as a good predictor for the correctness of a decision regardless the experience of a student.

The findings regarding the research questions have several implications for empirical software engineering as well as for software engineering education. Students' self-rated experience is not a good predictor for their actual performance in model comprehension tasks. Hence, the common approach to use self-rated experience as a selector for splitting participants into treatment and control groups or as a selector for experiment participation seems to be unreliable and should, therefore, not be used when it comes to experiments in the field of conceptual modeling and model-based development. Obviously, the same holds true for teaching conceptual modeling as students' perception on how experienced they are cannot be relied on. We will elaborate on the inferences that can be drawn from the results of the observational study in more detail, after discussing the major threats to validity of the investigation.

7.2 Threats to Validity

To address threats to validity of the observational study and the individual experiments, which exist for this type of study (cf. Campbell and Stanley (1963); Zhang et al. (2014)), we have employed certain strategies, which we will discuss in this section. In addition, we will discuss the remaining threats to validity of the investigation with respect to the inferences that can be drawn. In the next subsections we will place emphasis on the threats for the individual experiments as these are the most substantial. Nevertheless, a threat regarding the overall setting of the observational study remains. Namely, it cannot be ruled out that the aggregation of data from four distinct sets bears some risks. However, we did not detect considerable deviations of the individual experiment data from the aggregated one. This can be seen from the descriptive statistics for each individual experiment in Section 4. The data shows that results regarding the research questions are in line between the four experiments. By aggregating these data, we are confident that we can make more substantiated claims regarding the generalizability of the findings for each research question.

7.2.1 Threats to Internal Validity

Regarding the internal validity of the experiments two major threats must be discussed.

First, there is a threat from students either not seriously participating or trying to overperform. This would corrupt the measurement of performance. To avoid such threats, we designed the individual experiments as online questionnaires to be conducted within 20-30 minutes and gave the students a time frame of 5 days to participate. Thus, we assume that the duration of the experiments was adequate to ensure students did not lose interest during participation and that internal threats to history, maturation, or mortality do not apply. Since volunteers may bias the results because they are generally more motivated than the average student, we decided to conduct the experiments as a mandatory part of our requirements engineering courses and explicitly decided to give no bonuses or credits as motivation. Therefore, the experiments were designed to also serve as teaching material, achieving a learning effect on model perception. This was supported by extensive debriefings in class. The experimental setup was carefully adopted to meet national laws as well as comply with university's ethics regulations on student participation in software engineering experiments.

Second, there is a threat of participants not giving correct estimations of their experience. We tried to mitigate this threat by ensuring anonymous participation by the students. We assume that students do not have a need to provide incorrect answers if they are aware that the instructor cannot match the answers to individual students.

7.2.2 Threats to Construct Validity

The fact that our experiments use mandatory student participation without student grading or other bonus forms aids in avoiding threats from evaluation apprehension. As we use a minimum of teaching related to the experiment upfront and we do not use upfront briefings, we also lower threats from hypothesis guessing by using naïve subjects. In addition, we only use quantitative measurements. Reliability of our self-defined measurement for experience was controlled using Cronbach's alpha. Cronbach's alpha shows that the single items that form the construct of self-rated experience are not contradictory, i.e., that they might actually measure the same construct. However, the threat remains that the items shown in Fig. 1 might not be adequate to actually measure students experience. Particularly, it must be questioned whether students are capable of correctly estimating their experience. However, in this paper, we do not want to investigate whether self-rated experience is an adequate measurement for experience but whether self-rated experience is a good predictor for performance.

The example specifications were carefully adopted in close collaboration with industry experts to fit the experiment setup. In addition, we used pretest groups to validate the individual experiment setups and materials. Furthermore, the individual experiments were designed based on commonly reported experiment designs for manual validation tasks.

The comparability of the graduate and undergraduate results for each individual experiment could be harmed if the same participants would participate as graduate and as undergraduate students. To avoid such effects, we conducted each experiment first with graduate students, and then with undergraduate students afterwards.

An important threat remains regarding the measurements used for experience, confidence, and performance. Regarding experience and confidence, we cannot ensure that our measurements are indeed good measurements for experience and confidence of the participants. Self-rated experience and self-rated confidence are subjective measurements where each participant interprets the scales individually. Additionally, participants may lack the ability to self-assess. Thus, there is a risk that self-rated experience does not actually measure the experience of the participants. Therefore, we refrain from concluding whether self-rated experience is a good measurement for experience and whether experience is a good predictor for performance. However, we feel confident to investigate whether self-rated experience is a good predictor for student performance in model comprehension tasks. The performance measurement is defined as the percentage of correct answers as is commonly used in educational settings (see Section 3.3).

Since we want to generalize from undergraduate and graduate students to the level of skills and experience, we need to assure that graduate students can be considered more skilled and more experienced. For example, undergraduate students could also have work experience, or already hold another degree.

7.2.3 Threats to External Validity

Threats to external validity in software engineering experiments typically stem from the question whether the experiment results are generalizable to an industrial setting (Höst et al. 2000). While this is obviously not the case this time, it is also interesting to discuss whether the results indicate bad student self-perception when it comes to model-based engineering or if this is a general problem in model-based engineering. Experimental research in software engineering has often concluded that there are only minor differences between professional software developers and graduate software engineering students. In addition, our findings indicate that there is no difference between the different experienced student participants regarding their self-perception.

While we designed the experiment material in close collaboration with industry professionals to adequately represent industrial situations, we cannot rule out the possibility that students might be more precise in their self-estimation when it comes to academic tasks that are commonly used in university education. Also we must mention that students were asked to rate themselves on a 5-point semantic differential scale ranging from "very experienced" to "very inexperienced" and results could be different if they instead had been asked to rate themselves compared to their fellow students.

Lastly, we must mention that the results might not be generalizable to all conceptual modeling languages but might rather be limited to the modeling languages used in the experiments (i.e., ITU message sequence charts, functional design languages and interface automata). However, these modeling languages are comparable to a wide variety of commonly used languages (such as sequence diagrams, finite state machines, data flow diagrams, etc.). Therefore, we assume that the found effects hold at least for a large subset of conceptual modeling languages.

7.2.4 Threats to Conclusion Validity

We used expert reviews of the experiment material and we used pretests to validate students' ability to understand the material in the intended way. Note that pretest participants were not chosen from the courses the participants were recruited in. Furthermore, we involved industry professionals to ensure that the transformation of industrial examples into experiment material did not corrupt the principle intention of industrial problems and examples.

Regarding the conclusions about the different measurements, we must state that self-rated experience was measured on a personal level (i.e., how experienced does a person find themselves) and confidence on a task level (i.e., how confident is a person they conducted a certain task correctly). Therefore, we cannot conclude that confidence is a better predictor than experience but only that self-rated confidence measured on a task level seems to be a better predictor for performance than self-rated experience measured on a personal level.

7.3 Inferences

Findings from the experiments show that software engineering students' self-assessment of their experience is not correlated to their actual performance when it comes to perceiving and interpreting conceptual models. This implies that the common use of self-rated experience in empirical software engineering as well as in software engineering education must not be considered a good predictor for students' actual performance in model comprehension tasks. In consequence, there is a need for a reliable measurement of students' experience to serve as a control and selector variable in empirical software engineering and to fairly group students in educational settings. However, as other researchers came to different results for self-assessment of students' programming experience (e.g., Sillito et al. (2008), Feigenspan et al. (2012)), this result can obviously not hold true for all areas of software engineering research and education. But it must be considered a valid counter example for the use of students' self-rated experience, as it might not be restricted to conceptual models and model-based development alone but might be also transferable to other non-programming related areas of software engineering.

Our results show a considerable difference in students' self-perception compared to other research concerning programming, consequently our findings suggest that programming and conceptual modeling have to be treated differently in an educational setting as well as in empirical research. Our findings add to the body of research on teaching conceptual modeling in software engineering in so far as our findings show that there is a difference between learning programming and conceptual modeling which will most likely need to be reflected in different educational approaches for programming and modeling in software engineering. In addition, it must be considered that this difference might also exists for other areas of software engineering that might either be similar to model-based development or different from both programming and model-based development. As software education research so far mainly emphasizes programming, we assume there is further need to investigate other areas of software engineering as well.

Lastly, it must be considered that the identified problem situation with bad self-perception and self-reflection might not be specific to students when it comes to conceptual models. This might indicate that when it comes to conceptual modeling, people in general have a hard time rating themselves accurately, which should therefore also be the subject of further investigation.

8 Conclusion

In this paper, we reported an observational study investigating the results of four experiments conducted with a total of 368 student participants solving tasks regarding their perception of conceptual models. We compared the students' performance, self-rated experience, and confidence in decision making and distinguished between more advanced graduate students and less advanced undergraduate students.

We found that the use of self-rated experience as a commonly chosen measurement for participants' experience is not a good predictor for performance in model comprehension tasks. Since the use of self-rated experience is a common predictor for performance and thus used as a selector and control in empirical software engineering as well as in software engineering education, these findings show that there is a need for a more reliable measurement for students' experience.

Furthermore, we found that students seem to be aware whether an answer they are giving is likely to be correct or not. Therefore, self-rated confidence for a single task can indeed be used as a predictor for the correctness of this task.

Additionally, we differentiated between graduate and undergraduate students. While we showed that graduates do perform better and do rate their experience higher, our findings show no statistically significant differences between the two groups regarding the relation between self-rated experience and performance or the relation between confidence and correctness. Hence, graduate students must not be seen as having better self-assessment of their abilities.

Since our data comes from experiments conducted with students, the question remains if this is an issue pertaining only to students or if the problem is more wide spread. Therefore, future work needs to examine if the issue of unreliability of self-rated experiences as predictor for performance is limited to students or if a broader problem exists.

In summary, we found self-rated experience not to be a good predictor for performance in model comprehension tasks. This holds for undergraduates as well as for graduate students. In contrast, we found the degree program as measurement for experience and self-rated confidence measured on a task level to be more reliable predictors for student performance in model comprehension tasks.

Appendix

Further Descriptive Statistics

As some object to the treatment of semantic differential scale data as continuous data, we analyzed the data considering confidence and self-rated experience as ordinal data. Table 12 lists the observed variables. Instead of averaging correctness and self-rated experience, we consider the median values for these analyses. Table 13 presents the descriptive statistics. Note that performance is still measured on a ratio scale. Table 14 shows Spearman's Rho for all combinations of variables.

Table 12 Observed variables

Name	Type	Scale	Characteristic	Description
Correctness	Response	Ordinal	Correct incorrect	Students' answers are either correct or incorrect
Performance	Response	Ratio	0%-100%	The ratio of correctly answered questions
Confidence	Explanatory	Ordinal	1-5	The confidence a participant claims for giving an answers
Self-Rated experience	Explanatory	Ordinal	1-5	The median value of participants' self-rated experiences
Degree program	Explanatory	Nominal	Undergraduate graduate	Participants are either enrolled in anundergraduate or a graduate degree program

Hypothesis Testing (non-parametric)

As parametric tests cannot be used on ordinal data, we use non-parametric tests to evaluate the hypotheses. In particular, we use Mann–Whitney U tests to compare graduate students' confidence to undergraduate students' confidence, and graduate students' self-rated experience to undergraduate students self-rated experience. To compare students' confidence between correct and incorrect answers, we conducted a Wilcoxon signed-rank test. For examining the relationship between students' performance and their self-rated experiences, we evaluated non-parametric partial correlations. We compare graduate students' confidence when giving a correct answer to undergraduate students' confidence when giving a correct answer and graduate students' confidence when giving an incorrect answer to undergraduate students' confidence when giving an incorrect answer using Mann–Whitney U tests. We also evaluate non-parametric correlations between performance and self-rated experience for graduate and undergraduate students separately. Note that we do not

Table 13 Descriptive statistics

		N					Percentiles		
	Degree Program	Valid	Missing	Median	Minimum	Maximum	25	50	75
Confidence correct	Undergraduate	232	2	4.066[a]	1.0	5.0	3.469[b]	4.066	4.704
	Graduate	134	0	4.311[a]	1.0	5.0	3.557[b]	4.311	4.917
Confidence incorrect	Undergraduate	234	0	3.808[a]	1.0	5.0	3.169[b]	3.808	4.388
	Graduate	134	0	3.781[a]	1.0	5.0	2.845[b]	3.781	4.469
Performance	Undergraduate	234	0	50.00%	0.00%	87.50%	37.50%	50.00%	60.27%
	Graduate	134	0	58.33%	4.17%	91.67%	41.67%	58.33%	66.66%
Self-Rated experience	Undergraduate	227	7	2.461[a]	1.0	4.0	1.775[b]	2.461	3.028
	Graduate	132	2	2.658[a]	1.0	5.0	2.071[b]	2.658	3.329

a. Calculated from grouped data.

b. Percentiles are calculated from grouped data.

Table 14 Spearman's rho

		Confidence Overall	Confidence Correct	Confidence Incorrect	Performance	Self-Rated Experience
Confidence overall	Correlation Coefficient	1	.686**	.751**	0.050	.211**
	Sig. (2-tailed)		0.000	0.000	0.335	0.000
	N	368	366	368	368	359
Confidence correct	Correlation Coefficient	.686**	1	.675**	.257**	.210**
	Sig. (2-tailed)	0.000		0.000	0.000	0.000
	N	366	366	366	366	357
Confidence incorrect	Correlation Coefficient	.751**	.675**	1	.085	.238**
	Sig. (2-tailed)	0.000	0.000		0.105	0.000
	N	368	366	368	368	359
Performance	Correlation Coefficient	0.050	.257**	.085	1	.073
	Sig. (2-tailed)	0.335	0.000	0.105		0.168
	N	368	366	368	368	359
Self-Rated experience	Correlation Coefficient	.211**	.210**	.238**	.073	1
	Sig. (2-tailed)	0.000	0.000	0.000	0.168	
	N	359	357	359	359	359

**. Correlation is significant at the 0.01 level (2-tailed).

compare graduate students' and undergraduate students' performance again, as performance is measured on a ratio scale.

Hypotheses Tests Regarding Underlying Assumptions

Regarding the assumption that graduate students have higher confidence than undergraduates, our results show that on average, graduate students (Mdn = 3.756) are not significantly more confident than undergraduates (Mdn = 3.606), U =14570.00, z=-1.15, p = .248. Therefore, we **cannot reject HA**.2_0.

Regarding the assumption that graduate students self-rate their experience higher than undergraduates, we found out that on average, undergraduate students (Mdn = 2.461) rated their experience significantly lower than graduate students (Mdn = 2.658), U = 12598.50, z = -2.56, p = .011. Hence, we can **accept HA**.3_A.

Hypotheses Tests Regarding Confidence and Correctness (RQ1)

On average, confidence was higher when giving a correct answer (Mdn = 4.143) than when giving an incorrect answer (Mdn = 3.800). This difference was highly significant z = -7.054 < .001. Hence, we **accept H1**.1_A.

Hypotheses Tests Regarding Self-Rated Experience and Performance (RQ2)

A non-parametric partial correlation was conducted to evaluate the relationship between performance and self-rated experience while controlling for degree program. When we control degree program on the relationship between performance and self-rated experience, we find the following non-parametric partial correlation r_s = .05, p (one-tailed)= .178. This shows that independent of the degree program, better performing students do not rate their experience higher than weaker performing students. We therefore **cannot reject H2**.1_0.

Hypotheses Tests Regarding the Impact of the Degree Program on Confidence and Correctness (RQ3)

On average, confidence when giving a correct answer was higher for graduate students (Mdn = 4.311) than for undergraduate students (Mdn = 4.066). This difference was not significant U = 14079.50, z = -1.58, p = .114. Therefore, we **cannot reject H3**.1_0.

On average, confidence when giving an incorrect answer was lower for graduate students (Mdn = 3.781) than for undergraduate student (Mdn = 3.808). This difference was not significant U = 14972.00, z = -0.74, p = .457. Thus, we **cannot reject H3**.2_0.

Hypotheses Tests Regarding the Impact of the Degree Program on Self-Rated Experience and Performance (RQ4)

For undergraduates, Spearman's rho shows that performance for undergraduate students was not or at least only with a negligible effect (Rea 2014) correlated with self-rated experience r_s = .12, p (one-tailed) = .039. For graduate students, performance was not significantly correlated with self-rated experience r_s = -.10, p (one-tailed) = .120. Therefore, we **cannot reject H4**.1_0.

Acknowledgements This work was partially funded by the German Federal Ministry for Education and Research (BMBF) under grant no. 01IS16043V. We thank our Industry Partners for their support with the experiment materials used in the original experiments. Namely we thank Frank Houdek from Daimler, Peter Heidl and Jens Höfflinger from Bosch, and Stefan Beck and Arnaud Boyer from Airbus for their support.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Arisholm E, Gallis H, Dyba T, Sjoberg D (2007) Evaluating pair programming with respect to system complexity and programmer expertise. IEEE Trans Softw Eng 33(2):65–86. https://doi.org/10.1109/TSE.2007.17

Bastarrica MC, Simmonds J (2019) Gender differences in self and peer assessment in a software engineering capstone course. In: 2019 IEEE/ACM 2nd international workshop on gender equality in software engineering (GE), pp 29–32. https://doi.org/10.1109/GE.2019.00014

Berander P (2004) Using students as subjects in requirements prioritization. In: Proceedings. 2004 international symposium on empirical software engineering, 2004. ISESE '04, pp 167–176. https://doi.org/10.1109/ISESE.2004.1334904

Bergersen GR, Gustafsson JE (2011) Programming skill, knowledge, and working memory among professional software developers from an investment theory perspective. J Indiv Diff 32(4):201–209. https://doi.org/10.1027/1614-0001/a000052

Bergersen GR, Hannay JE, Sjoberg D, Dyba T, Karahasanovic A (2011) Inferring skill from tests of programming performance: combining time and quality. In: 2011 international symposium on empirical software engineering and measurement, pp 305–314. https://doi.org/10.1109/ESEM.2011.39, iSSN: 1938-6451

Bergersen GR, Sjøberg DIK, Dybå T (2014) Construction and validation of an instrument for measuring programming skill. IEEE Trans Softw Eng 40(12):1163–1184. https://doi.org/10.1109/TSE.2014.2348997

Biffl S (2003) Evaluating defect estimation models with major defects. J Syst Softw 65(1):13–29. https://doi.org/10.1016/S0164-1212(02)00025-0

Boud D, Falchikov N (1989) Quantitative studies of student self-assessment in higher education: a critical analysis of findings. High Educ 18(5):529–549

Bunse C (2006) Using patterns for the refinement and translationof UML models: a controlled experiment. Empir Softw Eng 11(2):227–267. https://doi.org/10.1007/s10664-006-6403-7

Byrne P, Lyons G (2001) The effect of student attributes on success in programming. SIGCSE Bull 33(3):49–52. https://doi.org/10.1145/507758.377467

Campbell DT, Stanley JC (1963) Experimental and quasi-experimental designs for research. Houghton Mifflin, Boston

Carver J, Jaccheri L, Morasca S, Shull F (2003) Issues in using students in empirical studies in software engineering education. In: Software metrics symposium, 2003. Proceedings. Ninth international, pp 239–249. https://doi.org/10.1109/METRIC.2003.1232471

Cen L, Ruta D, Powell L, Ng J (2015) Interaction driven composition of student groups for optimal groupwork learning performance. In: 2015 IEEE frontiers in education conference (FIE), pp 1–6. https://doi.org/10.1109/FIE.2015.7344266

Cook TD, Campbell DT (1979) Quasi-experimentation: design & analysis issues for field settings. Houghton Mifflin, Boston

Cushion CJ, Armour KM, Jones RL (2003) Coach education and continuing professional development: Experience and learning to coach. Quest 55(3):215–230. https://doi.org/10.1080/00336297.2003.10491800

Daun M, Brings J, Weyer T (2017) On the impact of the model-based representation of inconsistencies to manual reviews. In: Mayr H, Guizzardi G, Ma H, Pastor O (eds) Conceptual modeling. ER 2017. Lecture notes in computer science, vol 10650. Springer, Cham. https://doi.org/10.1007/978-3-319-69904-2_35

Daun M, Brings J, Krajinski L, Weyer T (2019a) On the benefits of using dedicated models in validation processes for behavioral specifications. In: Sutton Jr SM, Armbrust O, Hebig R (eds) Proceedings of the international conference on software and system processes, ICSSP 2019, Montreal, QC, Canada, May 25-26, 2019. IEEE / ACM, pp 44–53. https://doi.org/10.1109/ICSSP.2019.00016

Daun M, Weyer T, Pohl K (2019b) Improving manual reviews in function-centered engineering of embedded systems using a dedicated review model. Softw Syst Model 18(6):3421–3459. https://doi.org/10.1007/s10270-019-00723-2

Daun M, Brings J, Weyer T (2020) Do instance-level review diagrams support validation processes of cyber-physical system specifications: Results from a controlled experiment. In: Proceedings of the international conference on software and system processes, ICSSP 2020, Seoul, Republic of Korea, October 10-11, 2020. ACM, p 10. https://doi.org/10.1145/3379177.3388893

DeVellis RF (2017) Scale development. 4th ed. Los Angeles: Sage

Dick M, Postema M, Miller J (2001) Improving student performance in software engineering practice. In: Proceedings 14th conference on software engineering education and training. 'In search of a software engineering profession' (Cat. No.PR01059), pp 143–152. https://doi.org/10.1109/CSEE.2001.913835

Diedenhofen B, Musch J (2015) cocor: A comprehensive solution for the statistical comparison of correlations. PLOS ONE 10(4):1–12. https://doi.org/10.1371/journal.pone.0121945

El Emam K, Madhavji NH (1996) An instrument for measuring the success of the requirements engineering process in information systems development. Empir Softw Eng 1(3):201–240. https://doi.org/10.1007/BF00127446

Eskew RK, Faley RH (1988) Some determinants of student performance in the first college-Level financial accounting course. Account Rev 63(1):137–147

Falchikov N, Boud D (1989) Student self-Assessment in higher education: a meta-analysis. Rev Educ Res 59(4):395–430. https://doi.org/10.3102/00346543059004395

Feigenspan J, Kästner C, Liebig J, Apel S, Hanenberg S (2012) Measuring programming experience. In: 2012 20th IEEE international conference on program comprehension (ICPC), pp 73–82. https://doi.org/10.1109/ICPC.2012.6240511, iSSN: 1092-8138

Feldt R, Zimmermann T, Bergersen GR, Falessi D, Jedlitschka A, Juristo N, Münch J, Oivo M, Runeson P, Shepperd M, Sjøberg DIK, Turhan B (2018) Four commentaries on the use of students and professionals in empirical software engineering experiments. Empir Softw Eng 23(6):3801–3820. https://doi.org/10.1007/s10664-018-9655-0

Field A (2013) Discovering statistics using IBM SPSS statistics. 4th ed. London: Sage

Fucci D, Turhan B, Oivo M (2015) On the effects of programming and testing skills on external quality and productivity in a test-driven development context. In: EASE '15: proceedings of the 19th international conference on evaluation and assessment in software engineering. ACM. https://doi.org/10.1145/2745802.2745826

Goodwin CJ, Goodwin KA (2016) Research in psychology methods and design. John Wiley & Sons, Hoboken, NJ, USA

Hagan D, Markham S (2000) Does it help to have some programming experience before beginning a computing degree program? In: Proceedings of the 5th annual SIGCSE/SIGCUE conference on innovation and technology in computer science education (ITiCSE 2000). Association for Computing Machinery (ACM), pp 25–28

Hannay JE, Arisholm E, Engvik H, Sjoberg DI (2010) Effects of personality on pair programming. IEEE Trans Softw Eng 36(1):61–80. https://doi.org/10.1109/TSE.2009.41

Höst M, Regnell B, Wohlin C (2000) Using students as subjects-A comparative study of students and professionals in lead-Time impact assessment. Empir Softw Eng 5(3):201–214

ISO/IEC/IEEE (2010) International standard - Systems and software engineering – Vocabulary. In: ISO/IEC/IEEE 24765:2010(E), https://doi.org/10.1109/IEEESTD.2010.5733835

ITU (2016) International telecommunication union recommendation z.120: Message Sequence Chart (MSC). Tech. Rep. Z120, International Telecommunication Union

James T, Galster M, Blincoe K, Miller G (2017) What is the perception of female and male software professionals on performance, team dynamics and job satisfaction? Insights from the trenches. In: 2017 IEEE/ACM 39th international conference on software engineering: software engineering in practice track (ICSE-SEIP), pp 13–22. https://doi.org/10.1109/ICSE-SEIP.2017.31

Jamieson S (2004) Likert scales: How to (ab) use them? Med Educ 38(12):1217–1218

Jedlitschka A, Ciolkowski M, Pfahl D (2008) Reporting experiments in software engineering. In: Shull F, Singer J, Sjøberg DIK (eds). Springer, London, pp 201–228

Jensen LP (2015) Using consultation in student groups to improve development of team work skills amongst more reluctant students. In: Proceedings of the 43rd SEFI annual conference 2015 - diversity in engineering education: An opportunity to face the new trends of engineering, SEFI 2015

Jørgensen M, Teigen KH, Moløkken K (2004) Better sure than safe? Over-confidence in judgement based software development effort prediction intervals. J Syst Softw 70(1):79–93. 10.1016/S0164-1212(02)00160-7

Katira N, Williams L, Wiebe E, Miller C, Balik S, Gehringer E (2004) On understanding compatibility of student pair programmers. SIGCSE Bull 36(1):7–11. https://doi.org/10.1145/1028174.971307

Kirschner PA (1992) Epistemology, practical work and Academic skills in science education. Sci Educ 1(3):273–299. https://doi.org/10.1007/BF00430277

Kitchenham B, Pfleeger S, Pickard L, Jones P, Hoaglin D, El Emam K, Rosenberg J (2002) Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng 28(8):721–734. https://doi.org/10.1109/TSE.2002.1027796

Kumar AN (2008) The effect of using problem-solving software tutors on the self-confidence of female students. In: Proceedings of the 39th SIGCSE technical symposium on computer science education, SIGCSE '08. Association for Computing Machinery, New York, pp 523–527. 10.1145/1352135.1352309

Layman L, Williams L, Osborne J, Berenson S, Slaten K, Vouk M (2005) How and why collaborative software development impacts the software engineering course. In: Proceedings frontiers in education 35th annual conference, pp T4C–T4C. https://doi.org/10.1109/FIE.2005.1611964

Likert R (1932) A technique for the measurement of attitudes. Archives of psychology

Lumley T, Diehr P, Emerson S, Chen L (2002) The importance of the normality assumption in large public health data sets. Annu Rev Public Health 23(1):151–169. 10.1146/annurev.publhealth.23.100901.140546

Marshall L, Pieterse V, Thompson L, Venter MD (2016) Exploration of participation in student software engineering teams. ACM Trans Comput Educ 16(2, Article 5):38. https://doi.org/10.1145/2791396

Mcdowell C, Werner L, Bullock HE, Fernald J (2003) The impact of pair programming on student performance, perception and persistence. In: 25th international conference on software engineering, 2003. Proceedings, pp 602–607. https://doi.org/10.1109/ICSE.2003.1201243

Mishra T, Kumar D, Gupta S (2014) Mining students' data for prediction performance. In: 2014 Fourth international conference on advanced computing communication technologies, pp 255–262. https://doi.org/10.1109/ACCT.2014.105, iSSN: 2327-0659

Mkpojiogu EOC, Hussain A (2017) Assessing students' performance in software requirements engineering education using scoring rubrics. AIP Conf Proc 1891(1):020092. https://doi.org/10.1063/1.5005425

Müller MM (2004) Are reviews an alternative to pair programming? Empir Softw Eng 9(4):335–351. https://doi.org/10.1023/B:EMSE.0000039883.47173.39

Morgan PJ, Cleave-Hogg D (2002) Comparison between medical students' experience, confidence and competence. Med Educ 36(6):534–539. https://doi.org/10.1046/j.1365-2923.2002.01228.x

Newhall T, Meeden L, Danner A, Soni A, Ruiz F, Wicentowski R (2014) A support program for introductory cs courses that improves student performance and retains students from underrepresented groups. In: Proceedings of the 45th ACM technical symposium on computer science education, SIGCSE '14. Association for Computing Machinery, New York, pp 433–438. https://doi.org/10.1145/2538862.2538923

Nugroho A (2009) Level of detail in UML models and its impact on model comprehension: a controlled experiment. Inf Softw Technol 51(12):1670–1685. https://doi.org/10.1016/j.infsof.2009.04.007

Osgood CE, Suci GJ, Tannenbaum PH (1957) The measurement of meaning. Chicago: University of Illinois Press

Pinto G, Ferreira C, Souza C, Steinmacher I, Meirelles P (2019) Training software engineers using open-source software: the students' perspective. In: Proceedings of the 41st international conference on software engineering: software engineering education and training, ICSE-SEET '19, event-place: Montreal, Quebec, Canada. IEEE Press, Piscataway, pp 147–157. https://doi.org/10.1109/ICSE-SEET.2019.00024

Polančič G, Heričko M, Rozman I (2010) An empirical examination of application frameworks success based on technology acceptance model. J Syst Softw 83(4):574–584. https://doi.org/10.1016/j.jss.2009.10.036

Raza M, Faria JP, Salazar R (2017) Helping software engineering students analyzing their performance data: tool support in an educational environment. In: Proceedings of the 39th international conference on software engineering companion, ICSE-C '17. IEEE Press, Piscataway, pp 241–243. https://doi.org/10.1109/ICSE-C.2017.61, event-place: Buenos Aires, Argentina

Rea LM (2014) Designing and conducting survey research: a comprehensive guide fourth edition, 4th edn. Jossey-Bass, San Francisco

Rex K, Roth RM (1998) The relationship of computer experience and computer self-Efficacy to performance in introductory computer literacy courses. J Res Comput Educ 31(1):14–24. https://doi.org/10.1080/08886504.1998.10782238

Ricca F, Penta MD, Torchiano M, Tonella P, Ceccato M (2007) The role of experience and ability in comprehension tasks supported by UML stereotypes. In: 29Th international conference on software engineering, 2007. ICSE 2007, pp 375–384. https://doi.org/10.1109/ICSE.2007.86

Robson C (2016) Real world research. 4th ed. Chichester: Wiley

Runeson P (2003) Using students as experiment subjects – an analysis on graduate and freshmen student data. In: Proceedings 7th international conference on empirical assessment & evaluation in software engineering, pp 95–102

Salman I, Misirli AT, Juristo N (2015) Are students representatives of professionals in software engineering experiments? In: 2015 IEEE/ACM 37Th IEEE international conference on software engineering (ICSE), vol 1, pp 666–676. https://doi.org/10.1109/ICSE.2015.82

Sillito J, Murphy GC, De Volder K (2008) Asking and answering questions during a programming change task. IEEE Trans Softw Eng 34(4):434–451. https://doi.org/10.1109/TSE.2008.26

Sinha A, Smidts C (2006) An experimental evaluation of a higher-ordered-typed-functional specification-based test-generation technique. Empir Softw Eng 11(2):173–202. https://doi.org/10.1007/s10664-006-6401-9

Sjøberg DIK, Anda B, Arisholm E, Dybå T, Jorgensen M, Karahasanovic A, Koren EF, Vokac M (2002) Conducting realistic experiments in software engineering. In: Empirical software engineering, 2002. Proceedings. 2002. International Symposium n, pp 17–26. https://doi.org/10.1109/ISESE.2002.1166921

Sjøberg DIK, Anda B, Arisholm E, Dybå T, Jørgensen M, Karahasanović A, Vokáč M (2003) Challenges and recommendations when increasing the realism of controlled software engineering experiments. In: Conradi R, Wang AI (eds) Empirical methods and studies in software engineering: experiences from ESERNET, Lecture Notes in Computer Science. Springer, Berlin, pp 24–38

Tichy WF (2000) Hints for reviewing empirical work in software engineering. Empir Softw Eng 5(4):309–312

Venkatesh V, Bala H (2008) Technology acceptance model 3 and a research agenda on interventions. Decis Sci 39(2):273–315. https://doi.org/10.1111/j.1540-5915.2008.00192.x

Webb NM, Nemer KM, Chizhik AW, Sugrue B (1998) Equity issues in collaborative group assessment: group composition and performance. Am Educ Res J 35(4):607–651. https://doi.org/10.3102/00028312035004607

Wieringa R (2010) Design science methodology: Principles and practice. In: Proceedings of the 32nd ACM/IEEE international conference on software engineering, ICSE '10, vol 2. ACM, New York, pp 493–494. https://doi.org/10.1145/1810295.1810446

Witten IH, Frank E, Trigg LE, Hall MA, Holmes G, Cunningham SJ (1999) Weka: Practical machine learning tools and techniques with Java implementations. Working Paper, University of Waikato, Department of Computer Science, Hamilton, New Zealand, 99/11

Wohlin C, Runeson P, Höst M, Ohlsson M, Regnell B, Wesslén A (2000) Experimentation in software engineering: An introduction, Kluwer international series in software engineering, vol 6. Kluwer Academic, Boston

Zhang D, Fonseca P, Cuthbert L, Ketteridge S (2014) An investigation of the team knowledge and team performance of the Chinese engineering students in a senior technical module. In: 2014 IEEE frontiers in education conference (FIE) Proceedings, pp 1–8. https://doi.org/10.1109/FIE.2014.7044078, iSSN: 2377-634X

Marian Daun leads the research group "Cyber-physical Systems and Requirements Engineering" at paluno, the Ruhr Institute for Software Technology at the University of Duisburg-Essen. His research focuses on model-based software, systems, and requirements engineering for cyber-physical systems. He also conducts research on the use of empirical methods in software engineering and on improving teaching in this area. Marian Daun holds a holds a PhD in software engineering from the University of Duisburg-Essen and has been an author or co-author of more than 80 peer-reviewed journal, conference, and workshop publications. He has acted and acts as organizer, program committee member, and reviewer for renowned international scientific conferences and journals.

Jennifer Brings is working as an associate researcher at the University of Duisburg-Essen. Her main topic of research is model-based engineering of collaborative cyber-physical systems. She holds a master's degree in business information systems. Jennifer Brings studied at the University of Duisburg-Essen, Germany and at Ewha Womans University, Seoul, South Korea. She has served on program committees for international conferences and workshops and as reviewer for international journals.

Patricia Aluko Obe is a research assistant at the University of Duisburg-Essen where she studies computer science with a focus on software engineering. Her research is mainly focused on model-based engineering of cyber-physical system where she has co-authored peer reviewed papers published at reputable international conferences and journals.

Viktoria Stenkova worked as an associate researcher at the University of Duisburg-Essen with special interests in the area of requirements engineering for cyber-physical systems. She holds a bachelor's degree in computer science from the University of Duisburg-Essen. Currently, she is enrolled at the Ruhr-University Bochum where she finishes her master's studies.

Affiliations

Marian Daun[1] · **Jennifer Brings**[1] · **Patricia Aluko Obe**[1] · **Viktoria Stenkova**[1]

Jennifer Brings
jennifer.brings@paluno.uni-due.de

Patricia Aluko Obe
patricia.aluko-obe@paluno.uni-due.de

Viktoria Stenkova
viktoria.stenkova@paluno.uni-due.de

[1] University of Duisburg-Essen, Essen, Germany