On Integrating Student Empirical Software Engineering Studies with Research and Teaching Goals

Matthias Galster
University of Groningen
The Netherlands
mgalster@ieee.org

Dan Tofan
University of Groningen
The Netherlands
d.c.tofan@rug.nl

Paris Avgeriou
University of Groningen
The Netherlands
paris@cs.rug.nl

***Abstract*—Background: Many empirical software engineering studies use students as subjects and are conducted as part of university courses. Aim: We aim at reporting our experiences with using guidelines for integrating empirical studies with our research and teaching goals. Method: We document our experience from conducting three studies with graduate students in two software architecture courses. Results: Our results show some problems that we faced when following the guidelines and deviations we made from the original guidelines. Conclusions: Based on our results we propose recommendations for empirical software engineering studies that are integrated in university courses.**

Keywords-empirical software engineering; studies with students; teaching and research goals

I. INTRODUCTION

A. Problem

Empirical studies in software engineering are conducted to show strengths and weaknesses of products or processes, to investigate the feasibility of a new or existing approach, or to identify areas for improvement [1]. The type of the empirical study depends on the goals of the study, the resources available, and the constraints of the environment in which a study is conducted [1]. Study types include for example controlled experiments, quasi experiments, surveys or case studies [2, 3].

Recently, many empirical studies that use university students as subjects have been published. For example, a survey on controlled experiments in software engineering found that 87% of studies used students as subjects [4]. Students are usually easily accessible for academic researchers and therefore attractive subjects. Many of these studies with students are conducted as part of university courses. However, these studies are often viewed skeptically by researchers and practitioners. As argued by Carver et al., reviewers of scientific journals and conferences sometimes question the value of such studies [1]. This is mainly due to the following reasons [1]:

- The experience of students is not representative of software engineering professionals.
- Studies with students often use toy projects, rather than realistic industrial applications.

Both reasons contribute to external validity threats as they tend to limit the generalizability of research finding. On the other hand, the validity of studies should not be judged based on the use of students, but based on the goal of a study and if this goal justifies the use of students [5]. This means, studies with students can be useful to industrial and research communities if they are conducted in an adequate way [1]. To facilitate adequately conducted studies with students of university courses, guidelines for integrating empirical software engineering studies with teaching and research goals have been proposed by Carver et al. [1]. These guidelines try to balance the need for thoroughly conducted studies with the pedagogical (educational and teaching) goals of researchers and educators. However, detailed information on how to use these guidelines, or experience reports about studies that follow these guidelines are missing. Such experience reports and more detailed guidelines would help other researchers design and conduct studies as part of university courses.

B. Paper Goal

Similar to previous studies that present experiences with following guidelines for systematic literature reviews or experiences with guidelines for surveys in software engineering [6-9], we present our experience with integrating empirical studies with research and teaching goals. In particular, based on our application of the guidelines for integrating studies with research and teaching goals as proposed by Carver et al. [1], we have two goals:

- Collect and report experiences with balancing research and teaching goals in empirical software engineering studies.
- Discuss issues faced when conducting studies and report practical recommendations to other researchers.

The contribution of this paper is therefore a first-hand experience report on applying the guidelines proposed by Carver et al. This paper can be considered as complementary to Carver et al. who not only provide guidelines, but also describe considerations when conducting empirical software engineering studies with students, including a practical example. We believe that the findings reported in this paper can be beneficial for other researchers who conduct empirical studies as part of university courses. Note that the recommendations provided in this paper are based on our experience from conducting three studies, but are by no means backed up by more solid empirical evidence.

Please also note that we do not discuss the general use of students as subjects for empirical software engineering studies, or problems and validity threats that occur when using students as subjects in empirical studies.

C. *Paper Structure*

The remainder of this paper is organized as follows. In Section II, we discuss guidelines to balance teaching and research goals when conducting empirical software engineering studies. We will refer to these guidelines throughout the paper. In Section III, we provide an overview of the method we used when reporting our experiences. These experiences with applying the guidelines proposed by Carver et al. are discussed in Section IV. An overall discussion is presented in Section V before we conclude the paper in Section VI.

II. BALANCING RESEARCH AND TEACHING GOALS

In [1], Carver et al. present guidelines for researchers and educators for planning and conducting studies as part of university courses. The overall goal of these guidelines is to help conduct studies that provide value for research but also pedagogical value for students. These guidelines are based on the experience of conducting a large number of empirical studies in university courses in Italy, Norway, and the United States. As stressed by Carver et al., studies conducted as part of university courses should have a pedagogical value. Studies conducted during classroom hours or as homework consume a significant amount of the time allocated for a course. Therefore, in addition to researchers, students and instructors are stakeholders with various concerns in such studies. In particular, students are interested in what they can learn from participating in a study, while researchers are concerned about the quality of the data collected from students. Thus, a study within the context of a university course has to be carefully planned, executed and integrated into the course. Carver et al. pose the following nine requirements on such studies [1]:

R1: External validity issues must be consciously considered.
R2: The study must be properly integrated with the course.
R3: Ethical issues must be adequately addressed through the study design.
R4: The correct goal must be chosen for the study based on its environment.
R5: Study setting must be appropriate relative to its goals, the skills required and the activities under study.
R6: The effect of differences between the subject population and the target population must be discussed.
R7: Students should learn the value of using empirical studies to evaluate products and processes and how to conduct them so that they can later perform their own assessments.
R8: Group work or collaborative work should be included in the study.
R9: The study should include development projects where possible.

Based on these requirements, Carver et al. compiled guidelines in the form of a checklist that researchers can follow to balance pedagogical and research goals (Fig. 1). Items on this checklist are grouped based on when they become relevant during the process of planning and conducting a study: before the class begins, as soon as the class begins, when the study begins, and when the study is completed.

Checklist to balance pedagogical and research issues

1. Before the class begins

1.1 Ensure adequate integration of the study into the course topics
Requirements addressed: R1, R2, R4, R5, R7, R8 (?), R9 (?)

1.2 Integrate the study timeline with the course schedule
Requirements addressed: R1, R2, R5

1.3 Reuse artifacts and tools as appropriate
Requirements addressed: R1

1.4 Write up a protocol and have it reviewed
Requirements addressed: R1, R2, R3, R4, R5, R8 (?), R9 (?)

2. As soon as the class begins

2.1 Obtain subjects' permission for their participation in the study
Requirements addressed: R1, R3

2.2. Set subject expectations
Requirements addressed: R1, R3

3. When the study begins

3.1 Document information about the experimental context in detail
Requirements addressed: R1, R2, R4, R5, R6

3.2 Implement policies for controlling / monitoring the experimental variables
Requirements addressed: R1, R6

4. When the study is completed

4.1 Plan follow-up activities
Requirements addressed: R1, R2, R3, R6 (?), R7

4.2 Build or update a lab package
Requirements addressed: R1, R2, R3, R4, R6, R7, R8, R9

Figure 1. Checklist from the guidelines to balance pedagogical and research issues in empirical studies with students (based on [1]).

Please note that for checklist items 1.1, 1.4 and 4.1, some requirements are not clearly addressed, as indicated by "(?)". This means, only in some study designs these requirements are met. For details, please see [1].

III. METHODOLOGY

Our paper provides a retrospective view on how we conducted three empirical software engineering studies as part of two software architecture courses. Data that manifests our experiences were extracted from study protocols and research notes taken when conducting the studies, as well as during analysis of study results and study debriefings. When recording our experiences, we tried to follow preliminary guidelines for experience papers as proposed by Budgen and Zhang [10] to cover the content recommended for experience reports. This includes that we clarify our role (we are two senior researchers and one junior researcher directly involved in the three studies based on which we report our experience), describe our source of experience (see study descriptions in the following subsections) and present lessons learnt (see Section IV).

We report our experience from three studies conducted by the authors as part of two software architecture courses at the University of Groningen, the Netherlands, in 2010 and 2011. The goals of the software architecture course at the University of Groningen include learning the full architecture design lifecycle according to Hofmeister et al. (analysis, synthesis, evaluation) [11], as well as learning about architecture process, reuse, and knowledge. The theory is applied in an architecting group project of a non-trivial system.

All three studies were conducted as part of a seminar session that consisted of a lecture part and a practical assignment. Practical assignments were used to collect data, and were organized as individual assignments rather than group assignments. Some details of the three studies are presented in the following subsections. Please note that in-depth discussions of the technical details of the studies as well as their results are not relevant to this paper as we focus on the use of the guidelines for balancing teaching and research goals.

A. *Study 1*

Study 1 was an exploratory study about handling variability in software architecture. The research goal was to elicit problems and implications when handling variability in software architecture. The educational goal was to introduce students to variability and to give them hands-on experience in designing a system for variability. In total, 27 graduate students participated in the study. Subjects were given the description of a public transport system. Then, students were asked to perform a set of tasks related to the design of this system. As this was an exploratory study, all students gathered in one room. Data was collected through paper-based pre-questionnaires, post-questionnaires, and through worksheets that recorded work results delivered by students. The study has been published at the 9th Working IEEE/IFIP Conference on Software Architecture [12] in 2011.

B. *Study 2*

Study 2 was a controlled experiment to analyze different approaches for assigning weights to stakeholder concerns when making architectural decisions. The research goal was to analyze weighting techniques with regard to their impact on the output of a decision, the time required to perform the weighting approaches, their scalability, ease of use, learnability and attractiveness. The educational goal was to teach students about prioritizing architecture decisions, and to provide them with hands-on experience with weighting methods to prioritize architectural decisions. The study was linked to the architecting group project of the software architecture course about designing a smart home power save system. This means, the problem description as well as the tasks given to students during the study were related to the system that students designed as part of their architecting group project. However, the results of the tasks were not part of the group project deliverables. In total, 30 graduate students participated in the study. The experiment followed an in-between design. This means, all participants used the same weighting techniques, but in a different order. We created two groups in two separate rooms to control for the impact of the order in which techniques were applied. Giving the same tasks to all students ensured that all students got the same educational value. Data was collected through a paper-based post-questionnaire and worksheets filled in by students. The results of this study are currently under review for publication.

C. *Study 3*

Study 3 was an experiment to evaluate lightweight variability management. The research goal of the study was to analyze an approach to help decide what change to accommodate in a software architecture, in what order to implement this change, and to understand the modifications necessary to the architecture. The educational goal was to teach students about system and software evolution as well as to provide them with practical examples of how to evaluate change cases and their impact on the architecture. Similar as Study 2, this study was linked to the software architecting group project about designing a smart home power save system. Twenty-five graduate students participated in the study. We collected data through a paper-based post-questionnaire and worksheets. The study has not yet been published.

IV. Experiences and Lessons Learnt from Using Guidelines to Balance Research and Teaching Goals

In the following subsections we discuss the ten items on the checklist of Carver et al. For each item we provide a brief discussion on how we accommodated it in our studies. Furthermore, we discuss lessons learnt (including reflections on considerations by Carver et al.) and present some recommendations. Please note that we discuss some items on the checklist in more detail than others. This is because some items are not specific to empirical software engineering studies with students, but are relevant for empirical studies in general (e.g., writing and reviewing a protocol).

A. *Integrate Study into Course Topics*

Care should be taken to integrate studies with the topics of the course. If a study is "too focused on the goals of researchers, it can easily produce invalid results if students are not well prepared" or if the study is not related to the course [1]. Usually, course instructors and researchers would collaboratively set the goal of the study to align teaching and research objectives. In our three studies, researchers and instructors were the same persons. Thus, we were familiar with the course material and could easily determine how the studies fit into the course. For example, for Study 2 the educational goal was to give students hands-on experience with weighting methods for architecture concerns when evaluating architecture decisions. The research goal on the other hand was to compare two weighting methods.

All three studies included practical assignments to apply concepts introduced during a lecture. Therefore, by making data collection part of an assignment instead of giving students another comparable assignment to gain hands-on experience without data collection, we were able to integrate all our studies in the course. Interestingly, we noticed that

some students applied concepts used in the study in their final architecting group project reports (e.g., prioritized stakeholder concerns or change cases using techniques learnt during Study 2 and Study 3). Our lessons learnt include:

- We learnt that researchers and instructors being the same person reduces problems with communicating the pedagogical value to students and with motivating students to participate in the study. This was because researchers knew the course. Also, students knew (and trusted) the instructor / researchers and therefore felt less intimidated compared to a situation in which an external person conducts the study. On the other hand, researchers and instructors being the same person could make students feel subconscious pressure to participate in a study, or to provide answers that the researcher / teacher would find positive. However, we did not get the impression that students felt this kind of pressure but acknowledge that this could be an issue with less mature students, e.g., in undergraduate courses.
- As recommended by Carver et al., we found it useful to present the educational benefits to students before starting the study, already as part of a general introduction course in the first lecture of the software architecture courses. We did this in addition to a short statement of anticipated educational value at the beginning of each study.
- When balancing teaching and research goals, researchers should resist the temptation to introduce course topics simply to make the course fit the study goal, rather than the study goal being defined based on the course goals. For example, in the case of Study 2, the goal of the study and the study design evolved due to the need for integration with predefined course topics. We had to discard other study options due to the low level of integration with course topics. For instance, case study research could be an appropriate method to study weighting techniques, but would not have been applicable in our course setting. We believe that this is not a big problem for graduate courses where advanced software engineering topics are taught (and thus more flexibility with regard to course topics is possible). On the other hand, for undergraduate courses that should teach basic concepts, this could impose pedagogical problems if the course topics are adjusted according to study goals.

  Recommendation 1: To really meet teaching and educational goals we recommend that special attention is paid so that the study goal does not drive the teaching goal. This means, if the course content is prescribed by a curriculum, it should not be changed just to make a study fit in the course.
- Integrating a study into a course with a course project provides several benefits. First, it motivates students to participate in a study as topics discussed as part of studies can help students with their project work. Second, it allows instructors to check if students learnt something during the study by checking if concepts used in studies have been applied by students in their course projects. Third, it makes the study environment more realistic as course projects usually go beyond toy examples.

  Recommendation 2: To increase the learning experience and motivation of students to participate in studies, we recommend connecting studies to course projects in which concepts from studies can be applied. Here, it is important to clarify how study topics help with the project. This is in line with recommendations proposed by Carver et al.
- We found that the most crucial aspect in order to address external validity was to elicit the background of participants, and then to filter participants based on the goal of the study. For example, for Study 1 we excluded all students with no practical experience in order to make the subject population more similar to software engineering practitioners. Also, by teaching students about related concepts we ensured that participants had sufficient knowledge about the method under study. Furthermore, by teaching them concepts from industry (such as feature modeling in Study 1), we made students closer to professionals (see also Section IV.G about documenting the experimental context).
- We did not find any indication that it would be useful for students to include the topic of experimentation itself in the course curriculum (as suggested by Carver et al.). Only a very small number of students indicated interest in the experimental methodology itself. However, this could be specific to our studies and the interest of our students in learning about experimentation as a method to evaluate processes and methodologies.
- We (as researchers and instructors being the same persons) were able to obtain an understanding of the students as potential subjects while teaching the course to students. This helped judge whether the study goals were reasonable. In some cases, instructors might even find that their students are not mature enough or suitable to act as subjects in an empirical study.

  Recommendation 3: To increase study validity, we recommend that instructors judge the ability of their students to participate in a study.

B. Integrate Study Timeline with Course Schedule

Studies should be well integrated with the course schedule as students "must properly allocate their effort among various commitments" [1]. Schedule pressures might affect students' motivations. We integrated all three studies with the course schedule. Reminders about the study were sent to students a week before each study. For example, Study 2 was scheduled in the fourth week of the course after students had already taken some architecture decisions for their architecting group project. However, they still had to take more decisions and thus could benefit from an approach for weighting architecture concerns. This ensured that students were able to learn something from participating in

the study. Furthermore, it helped ensure that participants did not feel disconnected from the course when participating in the study. For Study 3, students had developed an initial architecture draft in their architecting group project and now had to work on an elaborated architecture, including system evolution. Thus, at the time the study was conducted they could benefit from an approach to evaluate future change in their architecting group project.

The course schedule of the software architecture course as taught in 2011 that integrated Study 2 and Study 3 is shown in Fig. 2. Fig. 2 also shows the deliverables of the architecting group project. Study 1 was conducted in 2010 and is therefore not shown in Fig. 2.

Lessons learnt include:

- We found it helpful to conduct a study later in the course when students acquired skills and background information on the topics taught in class. Thus, studies may not be introduced in a course too early as students might lack necessary knowledge to properly perform their assignments (except if lack of knowledge is desirable given the goal of a study). On the other hand, we found that the later a study is conducted in a course, the more likely students will skip it if they are busy with other courses, assignments and exam preparation. In busy times, students tend to finish their study assignments early in order to spend more time on studying for exams. This might affect the results of the assignment and thus the quality of study data. However, we believe that the time of the study depends on the goal of the study. For example, when studying the behavior of inexperienced architects or novice architects (such as our work in [13]), a study should be conducted before participants get exposed to architecting methodologies.

  Recommendation 4: To increase student motivation, it should be ensured that students are not busy with other tasks when scheduling the study. The overall workload of students (e.g., exams) should be taken into consideration.

  Recommendation 5: If a study has to be scheduled towards the end of a term when students are busy, it is helpful to keep a study short and simple. This could avoid compromising the goal of a study and the validity of data because students might carelessly perform tasks given to them during the study.

  Recommendation 6: When planning the time of the study, the goal of the study should be taken into consideration to avoid construct validity threats (e.g., measuring phenomena that students are not able to understand). If necessary, the study goal should be adjusted so that tasks are easy for students to complete.
- We scheduled all studies before the start of the course and presented the time of studies to students during the first lecture, together with a summary of the course. This was positive for students as they knew what to expect throughout the course and could plan the studies in their agenda. However, it imposed constraints on our studies: It reduced flexibility in adapting our research goals according to research results obtained between the start of the course and the time a study was conducted.
- We used post-questionnaires in all three studies to determine how students perceived the studies and if they felt that the studies helped them with their architecting group project. Information from post-questionnaires was valuable for instructors to evaluate the learning experience of students (see also Section V).

  Recommendation 7: To ensure a good learning experience and to find out how well the study fit into the course, checking the learning experience through post-questionnaires can help instructors gain insights into the effect of a study on student learning.

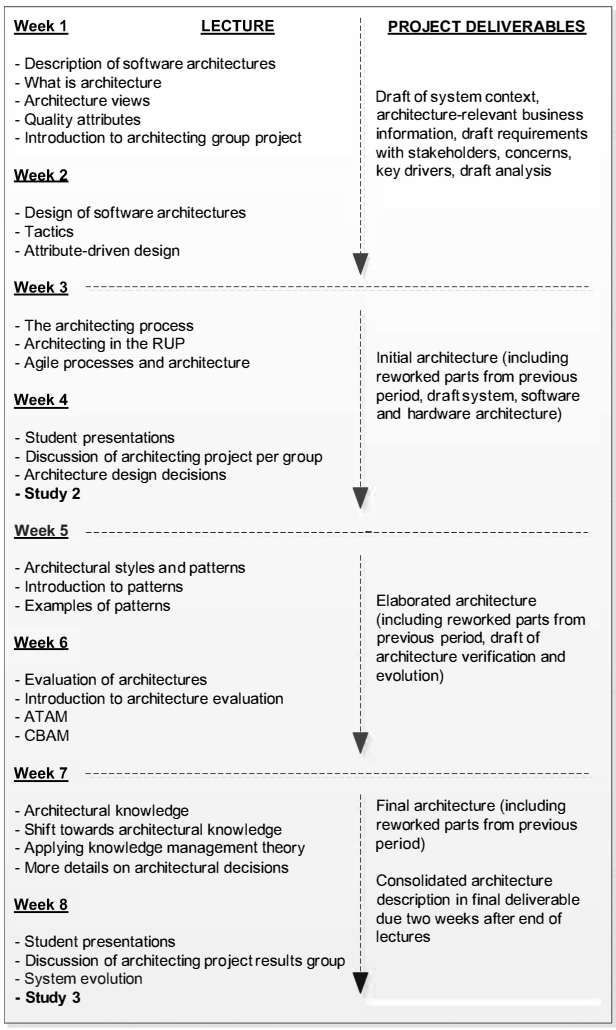

Figure 2. Example course schedule of the software architecture course.

- Overall, the scheduling of the study has a significant impact on how students perceive a study. According to our experience, students feel less as if they are

being tested or being put on the spot when the study fits well in the course schedule. This is because the study smoothly fits in the schedule and in the mindset of students provides a continuous learning experience.

C. *Reuse Artifacts and Tools as Appropriate*

To save time and to allow comparisons with other studies, researchers should search for existing artifacts and tools that can be reused in their studies [1]. For Study 1 we used a problem description that has been published for the ICSE Student Contest on Software Engineering (SCORE) 2011. Thus, we could consider this artifact of Study 1 as "tested". It also saved us some effort as we did not have to create a completely new requirements or system description. For Study 2 we identified tools to analyze decisions and for calculating weights based on the input from students. In Study 3 we found it helpful to "reuse" some concepts that we found in industry (e.g., influence matrices). This helped not so much with saving time or with comparing our study with others, but helped ensure that our study resembled the needs of industry (as stated by Carver et al.). This reduces threats to external validity. Lessons learnt include:

- We found it difficult to reuse existing experimental artifacts. However, this might depend on the study characteristics and study goals. Furthermore, we looked for reusable artifacts but it seems that current study artifacts are rarely made available to other researchers.
- Academic conferences, such as ICSE, could provide useful resources for designing studies with students. This is particularly true for reusable problem descriptions of software systems.

D. *Write up a Protocol and have it Reviewed*

For all studies we prepared a protocol that outlined the steps for the studies. We discussed the protocols among researchers and had them reviewed. We do not see this as a special requirement for integrating research with teaching goals but as a practice that every study should follow. Lessons learnt include:

- When reviewing a protocol for a study with students, attention should be paid not only to the soundness of the proposed research, but also to the pedagogical value of the study. However, most researchers and protocol reviewers tend to ignore this aspect and focus on research aspects.
  Recommendation 8: To increase the quality of protocol reviews, special instructions should be given to reviewers. This means, reviewers should be instructed to ensure that students receive adequate educational value from a study. On the other hand, pedagogical value and research quality should be balanced.
- We found that an external review is extremely useful when researchers and instructors are the same persons. This could particularly be helpful to get a judgment on the suitability of a study for the course.
- Our study protocols did not require a formal approval from an institutional review board (IRB) from our university. However, this depends on the institution of the researchers and where the studies are conducted. Nevertheless, even though no IRB approval was necessary, we obtained consent from students to participate in the study, made the study mandatory and allowed students to leave during the study. As there was no formal ethical review, we stressed the importance of paying attention to ethical issues to protocol reviewers.
- In all studies we assured students that the result of their assignment would have no influence on their grade. According to our experience, this encourages students to answer more openly and honestly without feeling forced to find the "correct" solution to a problem.
- When piloting student studies, we found it challenging to get appropriate candidates that would resemble the target sample (i.e., students of the particular software architecture course) well enough. This is because a) we did not want to use students from the actual sample as there was only a limited number of students enrolled in the course, and b) we wanted to avoid that details about the study were discussed between students who piloted the study and other students in the course. Thus, we used students from previous years who took the same course as well as students from another university who had not taken the course at our institution.

E. *Get Subjects' Permission for Participation in the Study*

Prior to studies, instructors and / or researchers should "inform students about the goals of the study", and "possible adverse consequences" as well as "measures taken to keep data anonymous" [1]. Therefore, before the beginning of each study, students were informed about the goals so that they could leave in case they did not feel comfortable in attending the studies. Moreover, students were given the opportunity to leave throughout the studies whenever they wanted. However, in our three studies no student left early.

In all three studies we did not have formal consent for participation. However, we emphasized that participation in the studies is optional and has no influence on the grade of students. Thus, by showing up for the study, participants implicitly expressed their permission for participation in the study. To keep data anonymous, ID's were given to students in Study 1 and Study 3. Lessons learnt include:

- Even though anonymous data was ensured in Study 1 and Study 3, for Study 2 we kept information about the architecting project group that students were in. This allowed us to give students detailed feedback about their work results in separate debriefing sessions (see also Section IV.I on follow-up activities). We checked with students beforehand and no objections had been raised about this procedure.
- We had to take into consideration that the number of participants showing up for the studies might have varied considerably, because of voluntary

participation. We had to accept the risk that studies might fail, because of a low number of participants, and plan for a repetition of the study in a different setting.

F. Set Subject Expectations

As the motivation of subjects is fundamental to valid studies, special care should be taken to explain to students what is expected from them. From the course schedule, students knew that all studies were limited in time (three hours). A few days before the studies, we sent a reminder to students, including a description about the topic and how it would fit into the course plan. Furthermore, at the beginning of each study, we presented the schedule for the study session. For example, for Study 3 the session started with an introduction to the topic of the session, followed by an overview of architecture evolution and evolvability and a discussion on anticipated change. Next, we presented a lightweight approach towards managing anticipated change and concluded the session with a practical assignment. This assignment was used for data collection. Additionally, we once again clarified that there would be no influence on the grade. We did not provide any incentives (such as money or gift cards) beyond the learning effect. As the studies were integrated in the course, students did not seem to expect to receive any monetary reward for participating in the studies. Lessons learnt include:

- We found it crucial to provide students with a goal of the study before starting the study to ensure motivation. When discussing the goal of the study, we paid special attention to avoid disclosing information that may bias the study.
  Recommendation 9: To set realistic expectations, researchers should think from the perspective of students when planning the study. However, this might be infeasible for some researchers. Thus, asking students from outside the sample provides an alternative view on a study goal. This could be done as part of the protocol review.

G. Document Experimental Context in Detail

To fully describe and critically appraise a study, it is necessary to record contextual information, including "specific characteristics and constraints that make the study environment unique" [1]. Thus, we documented the context of all studies. This included information about the subjects and their background (elicited through questionnaires), such as study program, previous degrees, practical experience, knowledge about software engineering and software architecture, etc. Furthermore, we documented the tasks given to students as well as their relation to course topics. We also recorded the information provided to participants. For example, in Study 1 participants received an introduction to variability modeling which then could be applied in their assignment. Lessons learnt include:

- It is very important to record all details about the study environment from the very beginning rather than in retrospective as important details might be missed. This is important in particular for experiments, especially if replications of the study are planned at other universities.
- Eliciting the background information is very important. It helps filter student results later on, depending on the goal of the study. For example, if the goal of the study is to gain insights into the behavior of real architects, the data of subjects with no industrial experience can be discarded. Furthermore, documenting the background allows for identifying the impact of industrial experience or education on the study results.
  Recommendation 10: We recommend the following items to be collected from study participants: 1) their degree obtained so far (in particular if graduate students are involved in a study); 2) the program of previous degrees; 3) the current program (if course is attended by students from different programs); 3) any industrial experience (years of experience, responsibilities, type of companies); 4) years of academic studies in software engineering and the topics of the study; 5) years of practical experience in software engineering and the topics of the study; 6) a self-assessment of how students rate their knowledge in software engineering and the topics of the study. Furthermore, we found it useful to document if exchange and international students were enrolled in our classes. Different educational systems may have an impact on the performance and responsiveness of students and therefore might affect study results.

H. Control / Monitor the Experimental Variables

As with any empirical study, factors that influence a study need to be controlled and monitored. Carver et al. argue that the same methods that are used in studies with practitioners can be used to collect different quantitative and qualitative measures during an empirical study with students (interviews, forms, etc.). Furthermore, evidence should be collected in a timely fashion, in a minimal-invasive fashion and considering that some data may be more sensitive than others.

In our studies we wanted each student to get similar educational value. Therefore, every student applied the same technique in the practical assignments. For example, in Study 2 and Study 3, all students used two weighting techniques. To control the experimental variables, we asked students to perform the tasks with the techniques in altering orders. This was to ensure that our controlled variables (techniques) provided similar educational value to all students.

Furthermore, instead of using an electronic study environment that would allow students to download task descriptions and upload task results we used paper-based data collection. This kept subjects focused on their tasks rather than being distracted by a technical environment (i.e., we ensured minimal invasive data collection).

As mentioned before, students were given questionnaires after completion of their practical assignment to check for treatment and to gather subjective feedback.

Carver et al. suggest group settings for conducting empirical studies. However, we did not use groups in our studies and therefore cannot report on if students are more comfortable with their classmates than with others. Lessons learnt include:

- In our studies we were not able to collect data automatically. As argued by Carver et al., automatically collected data may be more reliable than self-reported data. However, using automatic data collection might often not be possible due to logistic constraints. For example, in another study which did not follow Carver et al. we separately had to book computer labs weeks beforehand.
- We would find it useful to split item "Implement policies for controlling / monitoring the experimental variables" of the checklist in two parts: First, the quality of experimental variables should be checked. Second, it should be checked that all students get the same value from the study.

I. *Plan Follow-up Activities*

As argued by Carver et al., follow-up activities are often overlooked in empirical studies [1]. One significant part of empirical studies with students is therefore to plan for follow-up activities. In Study 1 we provided students with feedback on their results through a study summary. Based on these summaries, students were encouraged to comment on the study itself, but also on the topic of the study. In Study 2 we held six debriefings with students after the study was completed. These debriefings provided detailed feedback to students to increase the educational value of the study. During these meetings, we presented students the full study details and results. Moreover, we prepared individual packages, so that each student could see the impact of various weighting approaches on his / her architectural decisions. In each half-hour debriefing session, each student received printouts with his / her results and a researcher presented the results, and answered questions from students. Lessons learnt include:

- Detailed feedback through study summaries and through debriefing sessions was highly appreciated by students. We found that these feedbacks increase the learning experience of students.
- Providing feedback to students increases the willingness of students to participate in future classroom studies. Furthermore, it increases the interest of students in the research topic and potentially helps recruit students for internship and thesis projects.
- Organizing follow-up activities require significant time which might not be justified by additional research value. We found that study summaries require the least effort, while dedicated study debriefings seem most valuable for students.
- During debriefings, similar as in focus groups, researchers might learn about additional phenomena not considered during the study, or might find explanations for unexpected results in the data.
- Preparing follow-up activities and holding debriefings also allows the identification of additional threats to validity. For example, it could be checked if students really understood the tasks given during the study, or if they just gave random answers because they did not know on how to solve the practical assignment.

J. *Build or Update Lab Package*

For replication of an empirical study in either an educational or professional environment it is important to have a lab package that can be reused by other researchers. To build a lab package for Study 2 and to help other researchers replicate our study, we kept detailed notes on our experiences. In general, the lab package should also include mistakes that we made during our studies so that replications can avoid these. We do not have particular lessons learnt for this checklist item.

V. DISCUSSION

A. *Summary*

Based on our experience, we consider empirical studies with students as useful and valid research approaches. We also believe that the existence of guidelines for conducting such studies, and experiences from researchers that performed such studies can greatly strengthen the quality of future studies. Furthermore, we think that guidelines for studies with students reduce researcher bias. However, when conducting our studies, we felt the need for more detailed guidelines. For example, we would have benefited from course outlines of existing courses and how they integrated empirical studies. Such outlines should include timelines of integration. Also, a detailed checklist for documenting the context of studies would be useful. In Section IV we therefore presented a list of items that we think should be collected to document the background of students.

Overall, we find it challenging to design empirical software engineering studies with students that balance teaching and research goals. In particular, aligning the study goal and research questions with teaching goals is not always easy to do without compromising teaching goals. Planning a study that is aligned with teaching goals also involves more effort compared to studies that simply take students as subjects, without taking the course context into consideration.

As we did not offer incentives in terms of prizes or exam marks, we conclude that the main motivation for students to participate in the studies was the learning experience. One problem is that students who did not participate in the studies might have missed out on valuable learning experience. This means, some students might not receive the same pedagogical value as students who do participate in a study. However, as our studies were organized as seminars as part of the course, and the actual study in terms of data collection was done as part of a hands-on exercise, students might only have missed the exercise but would still have gotten insights from the instructional part of the study session.

We found it very useful to provide feedback to students and to check if students actually learnt something in the study. For all studies, we collected information about the learning experience and the study experience on post-questionnaires. For example, for Study 3, 72% of participants indicated that they believe that the study will help them with their architecting group project; 8% strongly believe that the study would help them with their architecting group project, and the rest were neutral about the impact of the study on their project. We also checked if students believe that they learnt something new from participating in our studies. For example, 68% of the participants of Study 2 believed that they learned something new, 8% strongly believed that they learnt something new, and 24% were neutral. We also asked if students enjoyed the assignment. For Study 2, 44% enjoyed the assignment, but 32% were neutral. In case of Study 3, 50% of the participants enjoyed the assignment. Furthermore, for Study 2 and Study 3 we also checked whether students applied concepts from the studies in their project reports and found groups who actually used the concepts from the study. Post-questionnaires also allowed feedback in form of open questions. This allowed students to comment for example on the quality of handouts, the problem descriptions given to students, or anything else they felt worth mentioning in the context of the study topic and how the study was conducted.

Strengths of using students included their availability, their knowledge in the topic under study (software architecture) and their motivation. However, motivation might differ when using undergraduate students. Weaknesses are that we found it sometimes difficult to generalize our results to practitioners. We believe relevance for practitioners is an important concern in the field of software architecture. We accommodated this weakness by filtering students during the data analysis based on their industrial experience. Even though this is possible for graduate students it might be more difficult to do with undergraduate students. More discussions on the use of students in software engineering research can be found in [14-16].

B. *Limitations of our Findings*

Based on [10], we discuss four types of limitations of experience reports:

- Construct validity: We did not employ any measure in our study and thus cannot discuss how well measures would have addressed the reporting of our experiences.
- Internal validity: We do not claim any causal link between observations and lessons, but simply provide recommendations based on our lessons learnt and our experience. Also, even though we have conducted more studies with students in the past, we did not discuss our experiences or lessons learnt from these studies. This is because we only reported on studies that followed recommendations by Carver et al.
- External validity: We only presented our experiences with three studies that followed the guidelines proposed by Carver et al. Therefore, we cannot claim that our observations and recommendations are applicable for other studies (in particular for studies that used other research methods than we did, e.g., case study research).
- Conclusion validity: The lessons learnt are purely based on our experience and do not have any empirical foundation. Furthermore, some lessons learnt and recommendations might be biased by our role as researchers and instructors as one person.

VI. CONCLUSIONS

We reported our experience in using guidelines for balancing teaching and research goals when conducting empirical software engineering studies with students. One of the main lessons we have learnt is to focus the study design itself on the course topic, rather than trying to make the course topics fit the study goal. Furthermore, we found reviewing protocols with regard to balancing study goals and teaching goals more challenging than reviewing study protocols that do not take teaching and pedagogical issues into consideration. As we are not aware of other papers that report experiences with guidelines for empirical studies with students, we cannot compare our experience with others.

In summary, we would appreciate more detailed guidelines about monitoring and controlling experimental variables in studies conducted as part of university courses. Also, example course outlines would help other researchers plan and schedule their studies properly. Finally, in order to explore lessons learnt further, future studies with students could be studied using a more thorough case study approach which explicitly collects data about using the guidelines of Carver et al. while planning and conducting such studies. This means, each study with students could be treated as a separate case in a case study about using the guidelines of Carver et al.

ACKNOWLEDGMENT

We thank all students who participated in our studies. This research has been partially sponsored by NWO SaS-LeG, contract no. 638.000.000.07N07.

REFERENCES

[1] J. C. Carver, L. Jaccheri, S. Morasca, and F. Shull, "A Checklist for Integrating Student Empirical Studies with Research and Teaching Goals," *Empirical Software Engineering*, vol. 15, pp. 35-59, February 2010.

[2] M. Zelkowitz and D. R. Wallace, "Experimental Models for Validating Technology," *IEEE Computer*, pp. 23-31, 1998.

[3] C. Wohlin, M. Hoest, and K. Henningsson, "Empirical Research Methods in Software Engineering," in Empirical Methods and Studies in Software Engineering, R. Conradi and A. I. Wang, Eds. Berlin / Heidelberg: Springer Verlag, 2003, pp. 7-23.

[4] D. Sjoberg, J. E. Hannay, O. Hansen, V. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal, "A Survey of Controlled Experiments in Software Engineering," *IEEE Transactions on Software Engineering*, vol. 31, pp. 733-753, 2005.

[5] W. Tichy, "Hints for Reviewing Empirical Work in Software Engineering," *Empirical Software Engineering*, vol. 5, pp. 309-312, 2000.

[6] M. Ciolkowski, O. Laitenberger, S. Vegas, and S. Biffl, "Practical Experiences in the Design and Conduct of Surveys in Empirical Software Engineering," in Empirical Methods and Studies in Software Engineering, R. Conradi and A. I. Wang, Eds. Berlin / Heidelberg: Springer Verlag, 2003, pp. 104-128.

[7] M. Riaz, M. Sulayman, N. Salleh, and E. Mendes, "Experiences Conducting Systematic Reviews from Novices' Perspective," in *Evaluation and Assessment in Software Engineering (EASE 10)* Keele University, UK: BCS, 2010, pp. 1-10.

[8] T. Dyba, T. Dingsoyr, and G. K. Hanssen, "Applying Systematic Reviews to Diverse Study Types: An Experience Report," in *International Symposium on Empirial Software Engineering and Measurement* Madrid, Spain: IEEE Computer Society, 2007, pp. 225-234.

[9] M. Staples and M. Niazi, "Experiences using systematic review guidelines," *Journal of Systems and Software,* vol. 80, pp. 1425-1437, September 2007.

[10] D. Budgen and C. Zhang, "Preliminary Reporting Guidelines for Experience Papers," in *13th International Conference on Evaluation and Assessment in Software Engineering (EASE)* Durham, UK: BCS, 2009, pp. 1-10.

[11] C. Hofmeister, P. Kruchten, R. L. Nord, H. Obbink, A. Ran, and P. America, "A General Model of Software Architecture Design Derived from Five Industrial Approaches," *Journal of Systems and Software,* vol. 80, pp. 106-126, January 2007.

[12] M. Galster and P. Avgeriou, "Handling Variability in Software Architecture: Problems and Implications," in *9th IEEE/IFIP Working Conference on Software Architecture* Boulder, CO: IEEE Computer Society, 2011, pp. 171-180.

[13] U. van Heesch and P. Avgeriou, "Naive Architecting - Understanding the Reasoning Process of Students - A Descriptive Survey," in *4th European Conference on Software Architecture* Copenhagen, Denmark: Springer Verlag, 2010, pp. 24-37.

[14] P. Berander, "Using Students as Subjects in Requirements Prioritization," in *International Symposium on Empirical Software Engineering* Rendono Beach, CA: IEEE Computer Society, 2004, pp. 167-176.

[15] M. Hoest, B. Regnell, and C. Wohlin, "Using Students as Subjects - A Comparative Study of Students and Professionals in Lead-Time Impact Assessment," *Empirical Software Engineering,* vol. 5, pp. 201-214, November 2000.

[16] M. Svahnberg, A. Aurum, and C. Wohlin, "Using Students as Subjects - An Empirical Evaluation," in *2nd International Symposium on Empirical Software Engineering and Management* Kaiserslautern, Germany: ACM, 2008, pp. 288-290.