Student-Centered Learning in a Large-Scale Engineering Course

Ola Leifler
Department of Computer and Information Science
Linköpings universitet
58183 Sweden
Email: ola.leifler@liu.se

***Abstract*---We describe an approach to teaching in an engineering course that intends to do three things without increasing the number of hours teachers spend: improve student activity, feedback and social skills, provide a deeper understanding of the topic of the course and make students more engaged in studying for an exam. In particular, we describe how we have transformed an existing traditional lecture-and-lab-based course into an active, cooperative learning course. Preliminary results indicate that students are much more active in studying for the final exam, and that their discussions help them to attain a much deeper understanding of concepts related to the course. ATTLS tests before and after the course show no significant changes in students' attitudes towards learning, however.**

Engineering courses in general are, in spite of extensive research in teaching and learning [1], often centered around two core activities: in-class lectures to define and explain theoretical topics of a course, and lab assignments where students solve a series of problems. In the scientific community studying learning and teaching, it is fairly well established that all forms of active learning methods are superior to passive modes of teaching [2]. However, faculty promoting ideas of more active learning styles in courses are often met by arguments from others that it may require too much effort to implement, or that students are not ready to take on the responsibility that comes with being active in learning. To test these assumptions in an engineering course, we took an existing course on object-oriented software design and applied techniques to improve student engagement in learning while ensuring that the teacher time spent was on par with previous years.

I. The course

The course was previously a traditional engineering course in computer science with respect to both content and form, with a set of lectures presenting textbook content, and a set of lab assignments individually graded by assistants. The course, though an advanced-level masters course, had not changed much with respect to either content nor form for the past ten years, despite research in the area. Students had expressed much appreciation for the course earlier, but in faculty evaluations, it turned out that students had not acquired relevant skills and abilities valued by professionals and others at the department. In the 2013 edition of the course, an attempt was made previous to introduce contemporary issues and other topics not in the course book through the use of lectures, which unfortunately failed to help students learn better and exam results dropped. When probing for the causes of this failure, we saw that students lacked good strategies for how to approach new subject matter, poor ability to reflect on the appropriateness of using the techniques they learned about, and no feedback on their progress in learning.

We saw that the deficiencies could partly be explained by the course format. Therefore, we decided to adopt a new approach to the subject, in which students' activities would be a more central part of the course. Part of the approach we adopted closely resembles the ``flipped classroom'' approach [3], which simply refers to having students actively solving problems in class, and demoting lecturing to be a passive learning activity students can choose to engage in outside the classroom. Although approaches to make students active in class is not fundamentally new, the introduction of online lectures means that educators can spend their time differently, and create quizzes and surveys in lectures to monitor student progress, and use that information to tweak educational activities in class[1]. Flipping the classroom has been demonstrated to have significant and positive effects on student activity in engineering classes in general [4], as well as in software engineering in particular [5]. Keeping the time budget for the course was paramount. The format changes introduced to the course were not meant to increase the time requirements for preparations, or total time spent during the course for the teaching staff.

First, we wanted to *improve student activity* during the course by replacing the lecture elements of the course through the use of a online lectures and in-class problem resolution. With the given time budget, and experiences from others using such flipped classrooms, we made the hypothesis that recorded, online lectures would be best used to demonstrate concrete, short examples, with supplementary reading of text books and code required to understand the topics in full. Some concepts seemed to be difficult for students to learn using online lectures, at least if they are the only format available [5]. To make students use different formats for learning about material in the course, we took the approach of only providing lectures for a selected subset of the material from previous

[1]For instance, using the Scalable Learning platform http://www.scalable-learning.com

years. From the original set of 18h of lecture material, we selected a set of 22 short segments, a total of 3.5h, that was considered most challenging for students, and best suited to make online lectures of. Together with supplementary demo material, an original course book and scientific papers to read on each major theme of the course, these online lectures became part of the core set of resources for students.

Second, to *make students engaged in studying and understanding concepts* in research papers and online lectures, we dedicated a set of seminars to theoretical topics and research in the field and wanted students to understand what kind of questions could be relevant to ask on this material. Allowing students to share questions on study material to others and answering others' questions has been shown to correlate positively with good exam results [6]. We wanted to help all students get better grades by understanding how to answer exam questions better through a contest in which students were to propose questions for the exam. It was made clear to them that, questions that met quality criteria *could* be included, but no guarantees were made that we would *only* use their questions, or that if we did, they would be identical in format or content to the ones they had originally formulated. In preparation for the theory-related seminars, students were to work individually to propose exam questions, and during the seminars, they were to work in their teams to select questions that best fulfilled the inclusion criteria put forth[2], so that their questions would be eligible for inclusion at the final exam. Students got feedback during the seminars on their understanding of the theoretical topics, and afterwards, on the quality of their proposed questions. All questions were posted online for others to see and comment on, and try to propose answer for.

Third, to *provide students with more feedback on their progress* during the course, and *improve their understanding* of the topic, we decided on a design for the lab series that focussed on student interactions in larger teams working on open problems in an existing open-source software project rather than complete, individual solutions to smaller problems. At the beginning of the course, we distributed all 140 students in teams of six, randomly assigned by a script. Each team consisted of three pairs, where each pair had a specific part of the code base to focus on for each task. To keep management efforts low, and all the while promoting interaction among students, we introduced a set of assignments that would require them to meet with one another before the deadline of each lab assignment and compare their results, and include a reflection of the differences in their reports. The rationale for this change was to make them aware of other possible solutions, and problems, when conducting the same general tasks. After the first two lab assignments, they would meet others who had studied the same code base but worked in other teams, to compare their results. After the third lab, they would have individual, oral examinations with their assistants to ensure that they had learned about concepts central to the lab course.

To summarize, Table I lists some of the qualitative changes of the course. Both the goals, the organization and the examination were changed to reflect a new *constructive alignment* [7] for the course. Students undertook realistic tasks with respect to the application of the techniques in the course, which was considered more important than merely applying a number of given techniques routinely in given contexts. In preparation for the written exam, the students´ teams proposed a total of 30 exam questions, and participated in more than 50 online discussions during the course. However, only the same number of students (30-50% of) attended the seminars in the revised edition of the course, the same number who attended lectures in the previous edition of the course.

II. Results

As the course had changed with respect to its' goals, we decided against validating our approach on exam results. Instead, we used student attitude surveys, data from the online platforms used on how many had watched the online videos and participated in discussions, and surveys with a number of students from the year before the change to the course. Students who had failed the previous exam were allowed to do parts of the new lab series instead of doing the exam again as the new lab series was supposed to evaluate the same knowledge and abilities as did the exam in previous editions of the course.

A. Interviews with previous year's students

Students who took the new lab series were unanimous in that they believed they learned more from studying existing software artifacts and trying to make sense of realistic code examples. Some examples:

> [Jag] kände att jag lärde mig mycket mer så här än att plugga inför tentor.
> ``I felt I learned much more this way compared to studying for exams."
>
> Jag tycker att detta sätt är mycket bättre än att tenta av en kurs eftersom vi fick sitta ner och prata om mönstren och lära sig av varandra än att sitta och stressa på en tenta!
> ``I think this assessment format is much better than to do a written exam since we got to sit down and talk about the patterns and learn from one another instead of stressing during a written exam!"

Several other students expressed that comparing results with peers and reflecting on the outcome helped them correct their misunderstandings. No students had expressed that the written exam was better at capturing their knowledge of design patterns.

B. Data from online activities

The online lecture platform gave us indications of how many students had watched the videos we posted, and done the quizzes for each topic. Here, we saw that almost all students had completed the module related to the lab assignments,

[2]such as being relevant for the topic and learning goals, that they should require proper understanding of course material, that they should be straightforward to grade and so on

	Individual learning (before)	**Collaborative learning (after)**
Task characteristics	Application of course-specific techniques on prepared problems	Conducting realistic tasks while trying to apply techniques from the course
Desired learning outcome	Ability to apply techniques in given contexts	Ability to reason about the applicability and consequences of techniques in various contexts
Interaction between students	Individual examinations, demonstrations to assistants	Reports submitted for others to see, group discussions after submissions.
Difficulties	Making students learn applicability, contexts and consequences	Ensuring that relate their activities to course goals, and meeting their expectations about what the course should be about.

TABLE I
Summary of changes

but only 25-30% had completed subsequent modules that were dedicated to the theoretical exam. In the mid-course evaluations, students were positive to watching lectures online, and no students made negative comments on the quality or usefulness of the online lectures. Almost all students who watched online videos also completed the quizzes.

C. ATTLS

One hypothesis with the revised course format was that students' attitudes to learning would change based on their experiences with peer discussions during seminar treatments of lab assignments and theoretical material. To measure changes in attitudes, we let students take an ATTLS ("Attitudes Towards Thinking and Learning") survey [8] before and after the course and compared the results. The response rates were 50% before the course, and 31% after the course. Both before and after the course, students expressed almost completely equal attitudes towards connected and separate learning, the two main learning attitudes differentiated by the ATTLS. No statistically significant changes could be detected between the two surveys.

III. Discussion

It seems that the online resources students did not feel they could use directly for their lab assignments were not well used. In general, students were not accustomed to participate in regular activities that did not award credits directly, but ``only'' prepared for the final exam. Thus, only the same number of students as last year came to class, and thus, student seminar groups tended to be formed ad-hoc. It was also apparent in the later theoretical seminars that some students were ill-prepared and lacked strategies to read scientific papers properly, which affected discussions negatively. This, too, could possibly be due to students' attitudes towards attending classes with no other direct incentives than being a preparation for an exam.

IV. Summary

To help students attain higher-level learning goals by solving engineering problems, it is imperative that they are able to do both concrete assignments with measurable results, and write meaningful reflections on their progress. Also, there have to be incentives for participation during a flipped classroom course, especially if group learning is to be successful. However, based on the same level of participation as in 2013, the course demonstrated a much higher activity among students.

References

[1] J. Handelsman, D. Ebert-May, R. Beichner, P. Bruns, A. Chang, R. DeHann, J. Gentle, S. Lauffner, J. Stewart, S. M. Tilghman, and W. B. Wood, ``Scientific teaching,'' *Science*, vol. 304, no. 5670, pp. 521–522, Apr. 2004.

[2] M. Prince, ``Does active learning work? a review of the research,'' *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, 2004.

[3] B. Tucker, ``The flipped classroom,'' *Education Next*, vol. 12, no. 1, pp. 82–83, 2012.

[4] G. Mason, T. Shuman, and K. Cook, ``Comparing the effectiveness of an inverted classroom to a traditional classroom in an upper-division engineering course,'' *IEEE Transactions on Education*, vol. 56, no. 4, pp. 430–435, Nov. 2013.

[5] G. C. Gannod, J. E. Burge, and M. T. Helmick, ``Using the inverted classroom to teach software engineering,'' in *Proceedings of the 30th international conference on Software engineering*, 2008, pp. 777–786.

[6] P. Denny, J. Hamer, A. Luxton-Reilly, and H. Purchase, ``Peerwise: students sharing their multiple choice questions,'' in *Proceedings of the Fourth International Workshop on Computing Education Research*, ser. ICER '08, New York, NY, USA: ACM, 2008, pp. 51–58.

[7] J. Biggs, ``Enhancing teaching through constructive alignment,'' *Higher Education*, vol. 32, no. 3, pp. 347–364, 1996.

[8] K. Galotti, B. Clinchy, K. Ainsworth, B. Lavin, and A. Mansfield, ``A new way of assessing ways of knowing: the attitudes toward thinking and learning survey (ATTLS),'' *Sex Roles*, vol. 40, no. 9-10, pp. 745–766, 1999.