THE ACTIVE-LEARNING TRANSFORMATION: A CASE STUDY IN SOFTWARE DEVELOPMENT AND SYSTEMS SOFTWARE COURSES*

Ross Sowell, Christopher Gill, Roger D. Chamberlain, Cindy Grimm, Kenneth J. Goldman
Department of Computer Science and Engineering
Campus Box 1045
Washington University
One Brookings Drive
St. Louis, MO 63130
314 935-6160
{rsowell, cdgill, roger, cmg, kjg}@cse.wustl.edu

Mark Tranel
Public Policy Research Center
University of Missouri - St. Louis
One University Blvd. 362 SSB
St. Louis, MO 63121
mtranel@umsl.edu

ABSTRACT

We present our experience in transforming a software development course and a systems software course from a traditional, lecture-based style to an active-learning format. We outline the common changes that were made in both courses, and provide a summary of the active-learning techniques that were successfully employed. We provide quantitative and qualitative evidence that this transformation was a success. In both courses, student grades and overall satisfaction with the course were increased with the transformation to active learning, despite teaching essentially the same curriculum.

INTRODUCTION

In the last decade, research in the learning sciences has repeatedly demonstrated the benefits of active learning, when teams of students are engaged in an ongoing process of inquiry and design, centered on real-world problems structured as an extended project [8, 9, 13, 19, 20, 21]. These research findings apply to all content areas, but they are particularly compelling with regard to science, math, and engineering design [5, 10, 11]. Over and over, researchers have demonstrated that the learning environments that result in maximum retention and increased ability to transfer learning to real-world settings are very different from the lecture-plus-problem-set paradigm traditionally followed in engineering programs [2]. However, this traditional, lecture-based style persists as the dominant form of instruction in computer science.

One barrier that inhibits the widespread adoption of the active-learning style in computer science is a lack of sufficient evidence that the techniques are effective in a wide variety of computer science courses. While there have been several papers in the literature with an emphasis on active-learning, the majority of these describe general active-learning techniques [4, 14, 15, 16, 17, 18, 22] or are focused on CS 1 and CS 2 [1, 3, 6, 7, 12]. This focus is justified, as those introductory courses are taught at most institutions, but there are also many common upper-level courses that to this point have been largely ignored in the literature. Another inhibitor to adoption of the active-learning style in more courses is the time and effort that it takes for an instructor to transform his or her course. In this paper, we present further evidence that active-learning can be successful in upper-level computer science courses and provide guidelines for performing the transformation.

We present a case study of two upper-level computer science courses (one systems software course and one software development course), taught by two different instructors, that underwent a similar transformation from a traditional, lecture-based style to the active-learning format. These two courses provide an interesting comparison because both are lab-based courses in which the concepts taught are relatively straightforward and the emphasis is placed on learning how to apply them. We present quantitative evidence that this transformation was successful in the form of student grades, exam scores, and course assessments, as well as qualitative data from student interviews. We also give a summary of the key techniques employed by the instructors during the active-learning sessions to serve as a model for others that desire to transform their own course.

THE ACTIVE-LEARNING TRANSFORMATION

The Courses Pre-Transformation

"Object-oriented Software Development Laboratory" (OOSDL) focuses on practical aspects of designing, implementing and debugging object-oriented software. It is a required course for all computer science majors, and serves as a technical elective for computer engineering majors. Topics covered by this course include developing, documenting, and testing representative applications using object-oriented frameworks and C++. Design and implementation are central themes to enable the construction of reusable, extensible, efficient, and maintainable software. Prior to the spring of 2009, this

course was taught in a traditional, lecture-based style, even though all of the assignments were laboratory assignments.

"Introduction to Systems Software" (ISS) examines the process whereby computer systems manage, interpret, and execute applications. It is a required course for all computer engineering majors and serves as a technical elective for computer science majors. This course covers fundamental algorithms for numerical computation, memory organization and access, storage allocation, and the sequencing and control of peripheral devices. Prior to the fall of 2008, this course was always taught in the traditional style. The scheduled contact time was devoted to lectures, and all assignments were to be completed outside of class. These consisted of both pencil and paper exercises as well as laboratory assignments that were hands-on with the machines.

Implementation of Active-Learning

OOSDL and ISS underwent the transformation to the active-learning format in the fall of 2008 and the spring of 2009, respectively. The primary changes that were common to both courses are as follows:

- Formal comprehensive in-class lectures were largely eliminated. Instead, short, informal lecturing and reviews are now done as needed during or interspersed with the active-learning sessions.
- The traditional lectures were recorded and made available to students online for preparation prior to class, or for review afterward.
- In-class time is now spent mainly solving problems or working on programming assignments in teams, and the instructors and teaching assistants make themselves available to provide help and guidance as needed.

There are minor differences between the two courses, including how groups are formed, how in-class work is graded, and the types of work that are assigned. These are detailed below.

The OOSDL class periods are now predominantly studio sessions, in which students work in teams on assigned programming exercises that explore different programming issues and C++ language features related to each course module. The professor and teaching assistants circulate throughout the studios to answer students' questions, offer suggestions, point out issues and nuances of the exercises on which the students are working, and otherwise serve as resources for the students as they work through the exercises. The studios are largely graded for participation. Students who submit a reasonably complete set of answers earn full credit.

During the ISS class periods, students are instructed to work on the currently pending written assignment or lab assignment in small groups of their choosing. The class meets either in the regular classroom or in a computer lab as appropriate for the task that class period. The professor and teaching assistant circulate and make themselves available during the class period to help out with the work the students are doing. Whatever they do not finish in class, they are expected to finish outside of class. Assignments are typically due one week after the in-class session that they are first presented. As a result of the above, students are being graded on the active work they are

doing in class. However, since it is not due immediately, they do have the opportunity to improve it outside of class.

Examples of the exercises used for active-learning in both courses are available online:
http://www.cse.wustl.edu/~rsowell/ActiveLearningExercisesOOSDL.html
http://www.cse.wustl.edu/~rsowell/ActiveLearningExercisesISS.html

RESULTS

Exam and Project Scores

In both courses, the midterm and final exams given after the active-learning transformation were based largely on the exams from the previous semester in an attempt to measure the same things. In ISS, the same large-scale final project was assigned both semesters. In OOSDL, the project score consisted of five or six lab assignments. One less lab assignment was made after the active-learning transformation, but as a whole, the laboratory assignments covered similar material during both semesters. The median, mean, and standard deviation of these scores are provided in Table 1.

The exam scores were significantly improved with the implementation of active-learning in all the exams, except for the OOSDL final exam. In this case the scores decreased, but this decrease was not significant. We also observed a decrease in the variability of all exam scores with the implementation of active-learning. We observed a similar increase in the project scores for ISS.

				Midterm			Final			Project		
Course		Style	n	Med	Avg	Dev	Med	Avg	Dev	Med	Avg	Dev
ISS	S2006	L	24	84.8	80.8	10.8	82.0	82.4	12.5	71.0	60.5	25.8
	F2008	A	26	90.0	88.3	8.9	93.0	91.0	6.2	82.0	79.8	17.5
OOSDL	F2008	L	29	79.0	76.3	16.8	90.0	85.6	11.3	97.0	95.4	7.6
	S2009	A	30	86.0	84.1	8.7	86.0	83.8	9.4	94.0	86.9	21.4

Table 1. Exam and project scores from the courses both before and after the active-learning transformation. "L" and "A" indicate whether the course was taught in the "Lecture" or "Active-learning" style, respectively. Exam scores increased in all cases except the OOSDL final, in which no significant difference was observed, while the variability of the exam scores decreased across the board. The project scores also increased with the implementation of active-learning in ISS.

The project scores for OOSDL decreased in the spring of 2009, but this decrease was partially influenced by the scores of one student that, for personal reasons, decided to take an Incomplete in the course. If we ignore the lab scores (many of which were zeros) for this student in the computation, then the mean becomes 89.19 with a standard deviation of 16.78.

Course Evaluations

Scores from the end of semester student evaluations are shown in Table 2. Among other things, the students were asked to give a score for the following three topics:

- Overall rating of course content.
- Overall rating for teaching quality.
- Overall satisfaction with the course.

			Evaluations		
Course	Term	Style	content	teaching	satisfaction
ISS	S2006	L	0.81	0.85	0.80
	F2008	A	0.83	0.88	0.84
OOSDL	F2008	L	0.77	0.83	0.74
	S2009	A	0.76	0.76	0.76

Table 2. Evaluation scores from the courses both before and after the active-learning transformation. "L" and "A" indicate whether the course was taught in the "Lecture" or "Active-learning" style, respectively. Overall student satisfaction with the course increased in both cases.

These scores were on a Likert scale from 1 to 7, with 1 being "poor" and 7 being "excellent". The questionnaire was changed in the fall of 2008, so that the scores were on a Likert scale from 1 to 9, with 1 being "poor" and 9 being "excellent". Therefore, we have normalized the scores in Table 2.

For ISS, the mean evaluation scores increased in every category with the implementation of active-learning. For OOSDL, the scores were slightly lower for course content and teaching quality, but the overall satisfaction with the course was still higher.

One student review of OOSDL was highly negative but the others were positive, and the reviews overall indicate that while not all students viewed the approach as positive, a significant majority of the responding students viewed it highly so.

Student Interviews

An external evaluator conducted student focus groups, one each of students from the fall 2008 session of OOSDL and ISS. A total of 17 students participated in the focus groups. The focus group protocol included three topics:
Topic 1: online lectures -- strength and weaknesses
Topic 2: use of classroom time -- how different
Topic 3: use of lab time -- strengths and weaknesses of working in groups

Key themes identified in the focus groups:

- Some students commented on the lack of real-time feedback from online lectures; cannot ask questions in the course of the lecture.
- Many students found it to be advantageous that they determined when to access the lecture material and that the lecture material was available for review. They found it to be an impediment to efficient use of the on-line material not having an index to the recorded lecture. If there was some specific information that they wish to go

to for review, it was just a random process of starting and stopping the lecture to find it.
- A number of students cited technical problems with accessing the online lecture material.
- An issue frequently mentioned was the new teaching approach required more hours to complete the coursework.

Instructor Reflections

The instructor for OOSDL reported that the students worked well in groups, and though initial concerns were expressed by some students, overall their level of engagement with the material appeared high and the studio discussions were active and effective. One key suggestion made to the teams was that during a studio, the students should rotate roles so that everyone had a chance to write code, debug code, and document answers for at least some of the exercises. An unexpected but welcome development was that one of the students often did parallel development of her own solution even when she was responsible for other things, just to have the experience of solving each exercise herself.

For future offerings of the course, the instructor for ISS is considering incorporating class work that is specifically not graded in any way other than participation. This might afford students the opportunity to try more creative approaches to a problem without the fear of being penalized for something that might not work.

CONCLUSION AND FUTURE WORK

We have presented our experience in transforming a software development course and a systems software course from a traditional, lecture-based style to the active-learning format. With this transformation, student grades and overall satisfaction with the courses were increased, despite teaching the same curriculum. Finally, we presented a summary of the key active-learning techniques employed by the instructors that made the transformation a success.

While the initial active-learning transformation was certainly successful, we see several opportunities based on our evaluation for further improvement of our approach:
- The recorded lectures should be indexed and enhanced with online notes.
- Eliminate even more of the formal lecture, and replace with short, informal lecturing when necessary during or just prior to the active-learning session.
- Effectively integrate peer review into the courses.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under CPATH Grant No. CNS-0722328. Any opinions, findings, and conclusions expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Beck, L. L., Chizhik, A. W., and McElroy, A. C., Cooperative learning techniques in CS 1: design and experimental evaluation, *SIGCSE Bull*., 37, (1), 470-474, 2005.

[2] Bransford, J. D., Brown, A. L., and Cocking, R. R., editors, *How people learn: brain, mind, experience, and school*, Washington, D.C.: National Academy Press, 2000.

[3] Briggs, T., Techniques for active learning in CS courses, *J. Comput. Small Coll*., 21, (2), 156-165, 2005.

[4] Chinn, D., Martin, K., and Spencer, C, Treisman workshops and student performance in CS. *SIGCSE Bull*., 39, (1), 203-207, 2007.

[5] Edelson, D. and Reiser, B., Making authentic practices accessible to learners: design challenges and strategies, In Sawyer, R. K., editor, *Cambridge handbook of the learning sciences*, New York, NY: Cambridge University Press, 2006.

[6] Gehringer, E. F. and Miller, C. S., Student-generated active-learning exercises, *SIGCSE Bull*., 41, (1), 81-85, 2009.

[7] Gonzalez, G., A systematic approach to active and cooperative learning in CS 1 and its effects on CS 2, In *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, 133-137, 2006.

[8] Hake, R., Interactive-engagement versus traditional methods: a six-thousand-student survey of mechanics test data for introductory physics courses, *Am. J. Phys*., 66, (1), 64-74, 1998.

[9] Johnson, D. R. and Smith, K., Cooperative learning returns to college: What evidence is there that it works? *Change*, (July/Aug), 26-35, 1998.

[10] Kolodner, J. L., Case-based reasoning, In Sawyer, R. K., editor, *Cambridge handbook of the learning sciences*, New York, NY: Cambridge University Press, 2006.

[11] Krajcik J. S., and Blumenfeld, P., Project-based learning, In Sawyer, R. K., editor, *Cambridge handbook of the learning sciences*, New York, NY: Cambridge University Press, 2006.

[12] Lau, K.-K., Active learning sheets for a beginner's course on reasoning about imperative programs, *SIGCSE Bull*., 39, (1), 198-202, 2007.

[13] Laws, P., Sokoloff, D., and Thornton, R., Promoting active learning using the results of physics education research, *UniServe Science News*, 1999.

[14] McConnell, J. J., Active and cooperative learning: tips and tricks (part I), *SIGCSE Bull*., 37, (2), 27-30, 2005.

[15] McConnell, J. J., Active and cooperative learning: more tips and tricks (part II), *SIGCSE Bull*., 37, (4), 34-38, 2005.

[16] McConnell, J. J., Active and cooperative learning: further tips and tricks (part III), *SIGCSE Bull*., 38, (2), 24-28, 2006.

[17] J McConnell, J. J., Active and cooperative learning: final tips and tricks (part IV), *SIGCSE Bull*., 38, (4), 25-28, 2006.

[18] Pollard, S. and Duvall, R. C., Everything I needed to know about teaching I learned in kindergarten: bringing elementary education techniques to undergraduate computer science classes, *SIGCSE Bull*., 38, (1), 224-228, 2006.

[19] Prince, M., Does active learning work? a review of the research, *Journal of Engineering Education*, 93, (2), 223-231, 2004.

[20] Redish, E., Saul, J., and Steinberg, R., On the effectiveness of active-engagement microcomputer-based laboratories, *American Journal of Physics*, 45-54, 1997.

[21] Sawyer, R. K., editor, *The Cambridge Handbook of the Learning Sciences*, New York, NY: Cambridge University Press, 2006.

[22] Schweitzer, D. and Brown, W., Interactive visualization for the active learning classroom, *SIGCSE Bull*., 39, (1), 208-212, 2007.