Personal Statement of Teaching – Sarah Heckman

Learning can occur anywhere, and a teacher’s responsibility is to facilitate learning inside and outside the classroom via lecture, activities, exercises, examinations, evaluation, and mentoring. My goal is to prepare software engineers for the workforce to solve the problems of tomorrow.

I regularly teach five of the ten undergraduate core courses in the area of software engineering and programming languages at NC State University. In a given week, I may give a lecture on the same topic at different depths in an introductory and upper-level course. By teaching courses at all levels in the curriculum, I have identified patterns of learning outcomes and have developed my teaching of those courses to acknowledge and promote connections. I frequently reference earlier coursework as foundational to a new concept and preview future learning that the students can look forward to in subsequent courses. As a graduate of our undergraduate program, I can share my journey with the students and serve as an example of completing the program.

My teaching philosophy is centered on engaging students with the material through active learning. Active learning is the foundation of teaching students discipline skills in software engineering and communication. I incorporate active learning into my own teaching scholarship through classroom investigations of new teaching and learning practices.

Active Learning

My teaching is centered on foundational lectures followed by active learning where students immediately apply concepts taught. I currently break up lectures with short three to five minute “pair and share” activities where students work together to solve a problem and then report their results via a Google form, as a substitute for clickers. Students benefit from using Google forms because there is a mechanism for assessment, which leads to engagement in the activity. Instructors benefit from Google forms because they can watch student responses as they are submitted, which leads to a discussion of misconceptions when the class reconvenes. Dr. Ed Gehringer and I received a DELTA IDEA grant in AY2011-2012 in support of developing additional instructor and student tooling to support the use of Google forms in the classroom. Most students reported that Google form exercises enhanced their learning and classroom engagement. While I largely feel that the current course delivery with in-class active learning exercises is effective in increasing student learning and engagement, I am interested in growing my teaching practices to incorporate other forms of active learning that will support deeper learning for students in my classroom, like the inverted classroom model.

Students have reported in class evaluations that they would like to engage more deeply with recently learned concepts and software engineering practices through larger programming exercises completed during class time. I plan on incorporating the inverted classroom model into several of my classes over the next several semesters with the goal of increasing student learning, efficacy, and engagement while supporting software engineering skills that may transfer to other assignments and courses. To support this goal, I attended the NC State’s Office of Faculty Development’s Summer Institute on the Scholarship of Teaching and Learning to develop a research study on the effectiveness of the inverted classroom. I will receive a grant from the Office of Faculty Development to support the evaluation of the course modification.

Software Engineering across the Curriculum

As a software engineer, I believe that active learning with appropriate emphasis on practices and processes can support software engineering skills and habits, increasing the likelihood of transfer of these skills and habits to completion of coursework and future careers. Some processes and practices are supported with tooling. I am incorporating Jenkins, a continuous integration server solution, to facilitate automated grading with code coverage and static analysis tools in three courses. The use of tooling supports discipline research on program analysis notifications and larger active learning lab activities in the classroom.
I am co-PI on an NSF grant with Dr. Emerson Murphy-Hill to increase the expressiveness and scalability of program analysis notifications. We have conducted research to understand how novice software engineers in CSC216 use or misuse notifications from code coverage tool, EcEmma, static analysis tool, FindBugs, and the Eclipse compiler. Notifications may lack rationale or support for deciding on the best course of action for resolution, which may hinder adoption. Studying novice developers in the classroom provides understanding into how tools can scaffold software engineering skills and provide the foundation for using similar tools when solving future software engineering problems.

My overall goal is to scaffold software engineering skills throughout the core courses in the software engineering and programming languages area to build student skills in requirements, design, implementation, and testing. Our students are very skilled at implementation. Testing is well integrated within our curriculum. Design and requirements are covered, but there is the opportunity to improve student learning in these areas.

Communication across the Curriculum

The Department of Computer Science has participated in research about how best to incorporate and integrate communication learning outcomes with the technical learning outcomes of our classes. The work was funded by a NSF grant titled CPATH II: Incorporating Communication Outcomes into the Computer Science Curriculum with PIs Mladen Vouk and Michael Carter. As part of this project, I have developed assignments that incorporate communication genres to support students learning of communication and technical skills. Several of the developed assignments require students to inspect requirements, write design proposals and rationales, and write and report the results of black box test plans, which support the idea of Software Engineering across the Curriculum especially in the areas of requirements, design, and test. While the CPATH project is ending, I will continue to require students to communicate in ways appropriate for computer scientists.

Scholarship of Teaching and Learning

A classroom generates many questions! Why did this assignment not work? Why are students not demonstrating learning of a concept? As I try new teaching techniques in the classroom, I want to evaluate their effectiveness to 1) see if I should continue using the teaching technique, 2) demonstrate to ABET continuous improvement of course delivery, and 3) contribute to the community of computer science educators through dissemination of the evaluation of classroom innovation. To achieve the third goal, I have moved my thinking from scholarly teaching to the scholarship of teaching and learning. I am active in the SIGCSE community by serving on the organization committee for the Technical Symposium on Computer Science Education. I have published in educational conferences. Where possible, I encourage the blending of discipline research with educational research by my Ph.D. student. Undergraduate students, as novice software engineers, can help us understand how learning in computer science occurs and better facilitate that learning through appropriate scaffolding and tool innovations.

Summary

My approach to education is to provide students the opportunity to engage with the materials that they must know to be successful. Student evaluations show that I am an effective communicator of computing concepts and help students with their understanding. I have won three student awards: the Most Receptive [Undergraduate] Professor Outside the Classroom in 2011 and 2013 and the Carol Miller Undergraduate Lecturer Award in 2014, named for a beloved retired lecturer in our department. Peer evaluations and continuing assessments for ABET show that students are meeting requirements for most learning outcomes in the classes I teach, with solid plans for improving student learning in areas of weakness. I have received educational grants from NSF and university entities to support my innovations in the classroom. As a teaching track faculty member at NC State, I have demonstrated ability as a highly effective educator of computer science with skills and resources for future distinction.