I construct learning environments that provide undergraduate and graduate students with opportunities for authentic practice in design, technology, and research. Such practice is often lacking in the classroom yet is crucial for preparing students to design and implement innovative solutions to social and technical problems upon graduation [4]. These opportunities for authentic practice help students develop deep subject-area expertise, but more importantly, explicitly support students to build *regulation skills*, i.e., cognitive, motivational, emotional, metacognitive, and strategic behaviors for reaching desired goals and outcomes [5, 6]. Regulation skills help students plan, execute, overcome challenges, and reflect on progress [3, 6]. Regulation plays a crucial role in student learning, but also in how students view themselves as learners. My goal is to design learning environments that support students growing over time, and throughout their lives.

In complex learning scenarios, students require significant training and support for developing their regulation skills. Mentors often lack the time and energy for effectively coaching more than a handful of students, and providing effective support through automated software is still infeasible [5]. To overcome these challenges, my students and I are designing organizational processes, community practices, and tools that promote socially supported regulation of learning (SSRL) [5, 6] so that students within a learning community can more effectively orchestrate their own learning, and so that teachers can sustainably train increased numbers of students in complex learning scenarios.

Design, Technology, and Research

I founded and direct Design, Technology, Research (DTR), a research program and learning environment that seeks to realize and develop undergraduate and graduate students' potential for developing novel technologies and creative solutions through design, engineering, and research.[1,2] Undergraduate and graduate students in DTR lead research projects in social and crowd computing, cyberlearning, human computer interaction, and artificial intelligence. In repeated 10 week-long studio sessions, students work with me and graduate student mentors to identify a research direction, explore and iterate over designs, prototype at varying fidelities, build working systems, conduct evaluative studies, and report findings through conference publications and workshops. The outputs are well-scoped, self-contained research projects that grow in complexity and generalizability over multiple studio sessions. This follows the spiral model of user-centered design and in agile development, which emphasize numerous iterations of testing interweaved with requirement gathering and technical development [8, 9].

DTR uses the Agile Research Studios (ARS) [10] model to scale SSRL within a research learning community.[3] It leverages multiple social structures to promote research progress and learning while scaling faculty time. Special Interest Group meetings (SIG meetings) bring together undergraduate students, graduate students, and faculty working on different projects in the same research area to plan work and devise strategies for overcoming challenges. Each SIG is its own mini-studio led by a faculty member who fades over time as a graduate student gains competencies in mentoring and becomes the leader of their own SIG. Studio meetings bring together all researchers in a studio to promote progress making, learning, and collaboration across SIGs. Students give and receive help through Pair Research [7] and share work through studio critique. In-person and online via Slack, students ask for help informally and receive support through their SIGs and from other students who have faced similar problems.

In four years, I trained 70 students (7 PhDs, 63 undergraduates) leading independent research projects through DTR. These students won 33 undergraduate research grants from the university,

[1] CTEC score (out of 6.0) for DTR: Overall Instruction Rating (5.6) and Overall Course Rating (5.7)
[2] `dtr.northwestern.edu`
[3] See my research statement for details on Agile Research Studios.

published 14 papers and extended abstracts at leading HCI conferences (CHI, UIST), and won 6 major ACM student research competitions, including at Grace Hopper and at ACM CHI in the past two years. 40% of DTR students are female; retention beyond their first quarter is at 94%, and most students continue until they graduate. My work on DTR was recognized through Murphy Society Awards for Advancing Undergraduate Engineering and an Office of the Provost award to advance associated learning technologies.

Pre- and post-test assessments show significant gains in student learning about design, technology, and research, increases in innovation self-efficacy [2], and positive feelings of belonging in a supportive community. Students noted that DTR helped them understand the spectrum of research processes, positively impacted their attitudes toward academic research, and led to significant shifts in their attitudes and beliefs in their ability to develop novel technologies and conduct STEM research. Students also became familiar with agile development and design research practices.

With the community structure for planning and support in place, I can orchestrate a research program with a few PhD students and 20 undergraduate students leading 10-15 research projects with 10-12 hours each week. I use: 4 hours for SIG meetings (four SIGs with 3 active projects and 5–6 students), 3 hours for the all studio meeting, and 4 hours for in-person and virtual office hours to respond to students on demand. Students have multiple opportunities for learning from mentors and peers in the studio each week, and are frequently exposed to research practices and products across projects and SIGs.

Figure 1: DTR group photo, Winter 2018.

In summary, DTR's structure and practices empower students to plan research work at weekly intervals and overcome challenges quickly with the support of peers and mentors. This allows students to conduct cutting-edge, independent STEM research along a faculty member's core research directions, as would be possible through dedicated 1-on-1 apprenticeship with faculty members but at just a fraction of the time required to support a much larger research learning community than would be traditionally feasible.

To further my educational mission, I founded the Agile Research University (ARU) program, which trains faculty to learn and adapt the ARS model for their use. The program brings faculty to visit the DTR studio over a three-day period, during which they observe ARS in practice. Visitors: (1) attend studio meetings and SIG meetings, (2) participate in pair research sessions, (3) meet with students, (4) learn about the ARS starter kit; (5) work with me to design their own studio and adapt tools/processes for their use. **ARU was successfully piloted last year with 8 current faculty members hosted from Berkeley, MIT, CMU, UCSD, Virginia Tech, Michigan, Northwestern, and NYU.** Continuing on the success of our pilot, I plan to diffuse the ARS model to a larger audience of 50+ faculty mentors and instructors through the Agile Research University program over the next 5 years.

Supporting SSRL in single quarter, project-based classrooms

While I believe that regulation skills are best developed over time through long-running programs such as DTR, many students' only exposure to open-ended problem-solving is through single quarter project-based classes. Given time and resource constraints, these classes often (1) lack the authenticity of professional work; (2) evaluate students on outcomes instead of process; (3) have students work in fixed teams and not with others outside their team; and (4) offer few opportunities for reflection and learning from failure. When developing a new seminar course, EECS 397/497: Social Computing & Crowdsourcing (2014, 2015, 2018),[4] for advanced undergraduate students and graduate students, and redesigning DSGN 401-2: Interaction Design (2015, 2016),[5] for masters students in the Engineering Design Innovation (EDI) program, I sought to address these shortcomings by (1) engaging students to work on real problems: students in EECS 397/497 developed their own research questions, and students in DSGN 401-2 consulted for startups in an accelerator program; (2) coaching students to reflect on their process and devise strategies for improving it; (3) creating numerous opportunities for students to collaborate with many others through hackathons, design challenges, design sprints, and short presentations; (4) setting milestones and using frequent studio critiques and check-ins to assess student learning, bring awareness to failures, and promote learning more effective strategies. These approaches are informed by known good practices and theories of cognitive apprenticeship [1]; students' assessments of their own learning in these classes suggest that these interventions helped them develop regulation skills with others in their class and helped them build core research and design skills.

Transforming large lectures into active learning environments

The rapidly increasing enrollment in Computer Science challenge teachers of traditional, lecture-based classes to engage large numbers of students and to challenge them to think actively and critically about problems. Students all too frequently become passive consumers of content. In helping to redesign *EECS 330: Human Computer Interaction* (2014),[6] and *EECS 101: An Introduction to Computer Science for Everyone* (2014, 2015)[7], I developed, within a flipped classroom model, in-class activities that use the diversity and scale of a large class as a means for enhancing learning. For example, in EECS 330, we used Google forms, spreadsheets, and app scripts to collect, aggregate, and visualize student responses to frequent design challenges in real time. In EECS 101, working with weekly guest lecturers, we devised activities that empower students to construct arguments and engage in debates about net neutrality, construct cryptographic schemes, and devise workflows for solving problems with a crowd. To promote students learning from each other, we made frequent use of *think-pair-share* so that students think on their own, construct knowledge with their peers, and then share their understanding with the class as a whole.

Delta Lab: An Interdisciplinary Research Lab and Design Studio

I co-direct the Delta Lab with Liz Gerber, Matt Easterday, and Nell O'Rourke.[8] Delta Lab is an interdisciplinary research lab and design studio with faculty and students from computer science, design, engineering, and learning science. Such interdisciplinary teamwork is critical in our efforts to study and design systems that fundamentally improve the way we design, work, learn, play, and interact. To create an environment of peer learning and mentoring, students present their work frequently throughout the research and design process. Everyone participates in pair research [7], a socio-technical system I helped develop that pairs students weekly to help one another on their projects to enhance productivity, collaboration, and informal learning.

[4] CTEC score (out of 6.0) for Social & Crowd Computing: Overall Instruction Rating (5.6) and Overall Course Rating (5.4)
[5] CTEC score (out of 6.0) for DSGN 401-2: Overall Instruction Rating (4.3) and Overall Course Rating (4.6)
[6] CTEC score (out of 6.0) for EECS 330: Overall Instruction Rating (4.8) and Overall Course Rating (4.6)
[7] CTEC score (out of 6.0) for EECS 101: Overall Instruction Rating (4.1) and Overall Course Rating (3.6)
[8] `delta.northwestern.edu`

References

[1] Allan Collins and Manu Kapur. Cognitive apprenticeship. In R Keith Sawyer, editor, *The Cambridge handbook of the learning sciences*, chapter 6. Cambridge University Press, 2005.

[2] Joanna C Dunlap. Problem-based learning and self-efficacy: How a capstone course prepares students for a profession. *Educational Technology Research and Development*, 53(1):65–83, 2005.

[3] Howard Gardner. Assessment in context: The alternative to standardized testing. In *Changing assessments*, pages 77–119. Springer, 1992.

[4] Elizabeth M Gerber, Jeanne Marie Olson, and Rebecca LD Komarek. Extracurricular design-based learning: Preparing students for careers in innovation. *International Journal of Engineering Education*, 28(2):317, 2012.

[5] Sanna Järvelä and Allyson F Hadwin. New frontiers: Regulating learning in cscl. *Educational Psychologist*, 48(1):25–39, 2013.

[6] Mariel Miller and Allyson Hadwin. Scripting and awareness tools for regulating collaborative learning: Changing the landscape of support in cscl. *Computers in Human Behavior*, 2015.

[7] Robert C Miller, Haoqi Zhang, Eric Gilbert, and Elizabeth Gerber. Pair research: matching people for collaboration, learning, and productivity. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 1043–1048. ACM, 2014.

[8] Don Norman. *The design of future things*. Basic books, 2009.

[9] Grant P Wiggins, Jay McTighe, Leslie J Kiernan, and Frank Frost. *Understanding by design*. Association for Supervision and Curriculum Development Alexandria, VA, 1998.

[10] Haoqi Zhang, Matthew W Easterday, Elizabeth M Gerber, Daniel Rees Lewis, and Leesha Maliakal. Agile research studios: Orchestrating communities of practice to advance research training. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 45–48. ACM, 2017.