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 [6]. 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 [8, 11]. Regulation skills help students plan, execute, overcome challenges, and reflect on progress [5, 11]. 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 help students view themselves and their learning positively, in ways that help them grow 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 [8]. To overcome these challenges, my students and I are designing organizational processes, community practices, and tools that promote socially supported regulation of learning (SSRL) [8, 11] 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 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 In DTR, undergraduate and graduate students 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, workshops, and blog posts. 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 interwoven with requirement gathering and technical development [13, 14].

DTR 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 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.

The DTR program has shown tremendous success over the last two years. During this time I hosted six studio sessions with 4 graduate students and 32 undergraduate students. 40% of DTR students are female; student retention beyond their first quarter is near 100%, and most students (plan to) continue until

1CTEC score (out of 6.0) for DTR: Overall Instruction Rating (5.7) and Overall Course Rating (5.9)
2dtr.northwestern.edu
they graduate. Students iteratively designed, built, and tested 18 new socio-technical systems. Multiple student papers have been accepted at conferences [2, 7, 1, 9, 10]. Three DTR students received 1st, 2nd, or 3rd at major ACM student research competitions (CHI 2015; Grace Hopper 2015). DTR students have applied for and received $30,000+ in funding for their projects; in the 2015–2016 academic year alone, 10 out of 11 DTR students applying for Undergraduate Research Grant funding received funding, an unprecedented success rate and the highest number of students supported by a single faculty member in any department across the University. Undergraduate students in DTR are applying to top PhD programs and receiving positions at the top tech companies such as Apple, Google, Microsoft, Yelp, LinkedIn, FitBit, and Vox Media. DTR received a Murphy Society Award from the McCormick School of Engineering in each of the last two years for its role in advancing undergraduate engineering education.

Pre- and post-test assessments collected each quarter show significant gains in student learning about design, technology, and research, increases in innovation self-efficacy [4], 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.

In qualitative interviews, students uniformly cited the mentoring, collaborative culture, and supportive community within DTR as key factors to their growth and development. For example, students commented that pair research [12] quickly resolved some of the most difficult problems by getting a fresh set of eyes on the problem and helped them gain skills that they couldn’t easily learn from tutorials. Students also commented on the value of having a cohort that supported each other.

With the community structure for planning and support in place, we can now orchestrate a research program with a few PhD students and 20 undergraduate students leading 10-15 research projects with less than 10 hours of faculty time each week.3 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 3 hours for in-person and virtual office hours to respond to students on demand. I can maintain awareness of progress across projects and still have time to respond to research challenges that I can best help students address. Students have multiple opportunities for learning from others in the studio each week, and are frequently exposed to research practices and products across projects and SIGs.

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.

3See my research statement for details on Agile Research Studios, a general model we are developing for scaling cognitive apprenticeship to advance undergraduate and graduate research training in STEM.
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 395/495: Social Computing & Crowdsourcing (2014, 2015), for advanced undergraduate students and graduate students, and redesigning DSGN 401-2: Interaction Design (2015, 2016), I sought to address these shortcomings by (1) engaging students to work on real problems: students in EECS 395/495 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 [3]; 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 101: An Introduction to Computer Science for Everyone (2014, 2015) and EECS 330: Human Computer Interaction (2014), 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 lectures, we devised activities that empower students 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 and Matt Easterday. 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 [12], 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 EECS 395/495: Overall Instruction Rating (5.7) and Overall Course Rating (5.6)
5 CTEC score (out of 6.0) for DSGN 401-2: Overall Instruction Rating (3.7) and Overall Course Rating (4.3)
6 CTEC score (out of 6.0) for EECS 101: Overall Instruction Rating (4.3) and Overall Course Rating (3.9)
7 CTEC score (out of 6.0) for EECS 330: Overall Instruction Rating (4.8) and Overall Course Rating (4.6)
8 delta.northwestern.edu
References


