Agile Research Studios: Orchestrating Communities of Practice to Advance Research Training

ABSTRACT
Undergraduate research experiences enhance learning and professional development, but providing effective and scalable research training is often limited by practical implementation and orchestration challenges. This paper introduces *Agile Research Studios* (ARS)—a socio-technical system that expands research training opportunities by supporting research communities of practice without increasing faculty mentoring resources. ARS integrates and advances professional best practices and organizational designs, principles for forming effective learning communities, and design of social technologies to overcome the orchestration challenge of one faculty researcher mentoring 20 or more students. We present the results of a two-year pilot of the Design, Technology, and Research (DTR) program, which used the ARS model to improve the quality of learning, produce research outcomes, and lower the barrier to participation while increasing the number of students who receive authentic research training.

Author Keywords
Agile research; community of practice; self-directed learning; regulation skills; socially shared regulation of learning; research training; socio-technical systems

ACM Classification Keywords
K.3.1. Computer Uses in Education: Collaborative learning

INTRODUCTION
Undergraduate research experiences provide numerous personal, professional, and societal benefits including enhancing student learning and broadening student participation and retention in diverse fields of study. Quality mentoring, sustained training, and practicing core modes of thinking in the discipline are crucial factors that contribute to student intellectual growth and interest in science and research careers [43, 48, 19, 44], especially among women and underrepresented minority students [32, 43, 33, 51].

Providing effective mentoring to undergraduate researchers is often limited by practical implementation and orchestration challenges [17]. 1-on-1 mentoring is effective but time-intensive [16]. As a research group expands in size, faculty have less time and attention to mentor each student. Without significant mentoring, undergraduate students have difficulty engaging in *authentic research* consisting of (a) core activities including designing a research plan, collecting and analyzing data, and preparing manuscripts, and (b) planning, monitoring and replanning research work. Without mentoring in core research skills, undergraduate students are usually relegated to rote activities such as data cleaning, transcription, or tagging [47], which is often less challenging and engaging. Some undergraduates perform more engaging and challenging activities, but may struggle to make consistent progress while waiting for busy mentors to help them with encountered obstacles [47]. These practical shortcomings can lead students to discount the value of research experiences and their own self-efficacy [19], resulting in fewer undergraduate students participating in research and few receiving the promised benefits of undergraduate research programs.

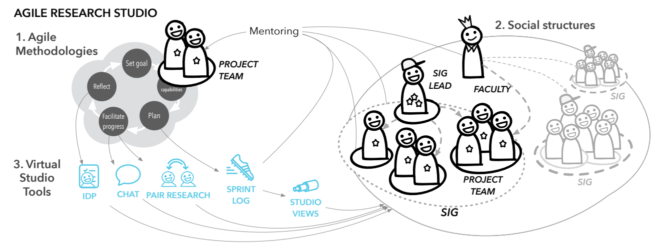

Figure 1. Agile Research Studios (ARS) is a new socio-technical model for creating a research community of practice that socially shares regulation of learning to apprentice undergraduate teams into research at scale. ARS methodologies, social structures, and tools help groups learn better together so more undergraduates can conduct authentic research.

The following question drives our research: **How might socio-technical systems train large numbers of students to conduct authentic research and produce research outcomes without increasing the orchestration burden on research mentors?** We propose *Agile Research Studios* (ARS), a new socio-technical model for research training consisting of processes, tools, and social structures for orchestrating research training within research communities of practice in which students collaborate to learn and conduct research and develop their abilities to be more self-directed [1, 49], see Figure 1. ARS (1) adapts agile processes [46, 13, 31] to research training so students learn to more effectively plan authentic research inquiry and (2) makes effective use of the expertise and resources across the research learning community to support research progress. We argue that by more fully leveraging the support of the research community, this approach will allow more students to engage in authentic research activities and produce research.

We study the ARS model through an exploratory case study of the Design, Technology, Research (DTR) program, a research learning environment implemented and led by a pre-tenure faculty researcher (the first author). DTR uses the ARS model to support a community of 21 student researchers including 18 undergraduates, 1 post-bac, and 2 PhDs with less than 12 hours of faculty time each week. Over two years, DTR hosted 36 students who led 18 research projects, published 8 papers and extended abstracts at peer-reviewed ACM and AAAI conferences, and won 3 awards at ACM student research competitions. Analysis of ARS activities show significant engagement in planning activities and students reported developing planning skills. Students helped one another regularly across project teams and reported shifts in their help-seeking and help-giving dispositions.

Our paper makes the following contributions to CSCW:

- *Agile Research Studios* (ARS), a socio-technical system for scaling research training that integrates effective work processes, social structures, and technologies to create sustainable research communities of practice by (1) adapting agile methodologies to research training and (2) making effective use of the expertise and resources across the community to respond to needs and support progress making.
- A case study of *Design, Technology and Research* (DTR), a research learning community that used the Agile Research Studios model to support a community of over 20 students researchers and one pre-tenure faculty mentor. Early results suggest ARS's efficacy in facilitating students' learning to plan research work and built a supportive community in which students regularly provide and receive help.

BACKGROUND

Training researchers is difficult because it requires students learning the processes, cultures, and mindsets of expert researchers [21]. A faculty member must balance the time and effort required for training students with the research group's need for productivity [18]. With limited time, faculty typically reserve 1-on-1 mentoring for a small number of well-prepared graduate students. Providing effective training to an increased number of students, including undergraduates who are less-prepared, thus requires scaling faculty time in ways that honor the research group's need for productivity.

Our work focuses on orchestrating the development of *regulation skills*, i.e., cognitive, motivational, emotional, metacognitive, and strategic behaviors for reaching desired goals and outcomes [30, 34]. Student researchers must develop regulation skills needed for conducting authentic research that demands (1) self-directed research planning, monitoring, reflection, and replanning [1]; and (2) adopting effective help-seeking and collaboration to overcome challenges [36, 34]. We argue that *developing regulation skills will permit student researchers quicker access to more central activities within the community of practice* that provide authentic research experiences that are more likely to lead to the desired learning and work outcomes.

Orchestration Challenges

Our work seeks to address practical socio-technical challenges in orchestrating the development of regulation skills, centered around research planning and getting help.

Research Planning

- **Doing all research steps.** In many research labs, even experienced undergraduate students may only perform a single research step (e.g., data collection); they rarely engage in all central tasks such as planning projects, generating and testing hypotheses, and authoring publications [47]. This allows less experienced undergraduates to participate in research, but limits their learning.

- **Doing planning.** As novice researchers, undergraduates lack skills for forming feasible and effective research plans, monitoring these plans as work progresses, and adjusting these plans in response to the inevitable challenges that arise in the course of research [1]. As a consequence, mentors often make themselves responsible for planning or tolerate projects going off track when they have too many students for whom to explicitly plan out work.

- **Learning planning.** When faculty prioritize short-term research productivity, they may use scarce 1-on-1 mentoring time predominantly for communicating work progress and overcoming technical problems. As a consequence, teaching self-directed planning and reflection is often deferred [47], even though these practices are vital for developing metacognitive skills [42, 10, 15].

Getting Help

- **Distributed help.** Without sufficient guidance, student researchers can easily become lost, confused, and frustrated [47]. Distributing help across both students and educators is difficult to encourage as it runs counter to common educational norms in which educators are the main source of help [23]. Students may waste a significant amount time before seeking help on research projects, and often only at a project's end [5]. While senior mentors are the most capable of addressing a wide variety of needs, their availability is most limited [25] and are seldom able to provide undergraduates researchers with all the help they need [47].

- **Scaffolding help-getting.** Even with a community of helpers available, getting help can be challenging for students because the community consists of many individuals working on different projects, making the task of identifying, selecting, and enlisting qualified helpers difficult [36].

- **Learning help-seeking.** Students often need additional help-seeking skills in order to learn effectively [36, 37, 41]. Without them, students may be reluctant to ask for help even when they need it and help is available [40]. In a survey of 123 university students undertaking student-led research projects, only 3% of students reported that getting help on research would support progress [5].

Models for Orchestrating Training

Traditionally, young researchers develop regulation skills through apprenticeships. Apprenticeship provides powerful model for research training but imposes an enormous orchestration burden on individual faculty mentors [17]. In the *1:1 apprenticeship* model, students work directly with expert researchers (Figure 2, Left). This form of training is effective

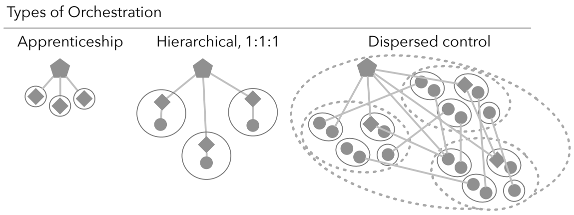

Figure 2. Only dispersed control and networked orchestration can overcome the "1:X" challenge to allow students to conduct authentic research at scale.

but labor-extensive, as it "...requires a very small teacher-to-learner ratio that is not realistic in the large educational systems of modern economies" [16]. As a consequence only small numbers of graduate students receive training. In the *hierarchical, 1:1:1* model, faculty mentors delegate undergraduate mentoring to graduate students (Figure 2, Middle). But graduate students themselves have yet to master research, let alone effective methods for effectively guiding others [45].

In order to overcome the shortcomings of 1:1 and 1:1:1 models, we must ultimately disperse control of orchestrating learning and support across a network of student researchers to address the "1:X" challenge of allowing one teacher to successfully respond to the learning needs of many students [2] (Figure 2, Right). In this model, research activities are situated within a learning community [6, 9]—a community of practice [50] designed for learning. Learning communities leverage the diversity of member expertise, value individual contributions, support continual advancement of knowledge and skills, emphasize learning how to learn, and provide mechanisms for sharing what is learned. We aim to apply the insights and design principles of learning communities to support research training for large numbers of student researchers in ways that respect faculty mentors' time and their need to make effective research progress.

A Socio-Technical Approach to Orchestration

To scale mentor time, our work aims to distribute planning and help activities to students and their peers by (a) developing students' research ability and self-directed learning skills so that they self-regulate learning (SRL) to conduct research more independently; (b) promoting student teams' socially shared regulation of learning (SSRL) so teams orchestrate their own learning; and (c) connecting students to peers, resources, and instruction to co-regulate learning (CoRL) so that students support each other beyond immediate project teams [30, 34, 1]. These activities should reduce the burden on mentors, but orchestrating them is itself time-consuming for mentors and difficult to support with software alone [30]. Our socio-technical solution—Agile Research Studios—consists of methods, community structures, and tools that collectively orchestrate research training so that such activities can feasibly take place within the community.

RESEARCH QUESTIONS AND DESIGN ARGUMENTS

How might we design socio-technical systems to train large numbers of students to conduct authentic research and produce research outcomes without increasing the orchestration burden on research mentors? Specifically, in this project we focus on planning and help seeking by asking:

RQ1: How can we orchestrate learning and support for student researchers to effectively plan authentic research inquiry, monitor and reflect on their progress, and make adjustments?

RQ2: How can we orchestrate help-seeking and collaboration to effectively leverage the distributed expertise of its members to support progress-making and learning?

We propose that we can address these questions by combining learning communities, agile methodologies, and online technologies to create Agile Research Studios (ARS) that supports students (1) learning self-directed research planning, monitoring, reflection, and replanning so that they can lead their own projects; and (2) adopting effective help-seeking and collaboration so that they support each other to learn and make progress. ARS consist of: (a) *agile methodologies*, (b) *social structures* including team meetings, special interest group meetings and studio meetings; and (c) *virtual studio tools* including sprint logs, resources, studio views, chat, pair research, research logs, and individual development plans.

ARS address the orchestration challenges of learning to plan research in the following ways:

- **Doing all research steps.** To engage more students in authentic research, ARS adapts agile methodologies to *slice* research work vertically to fit student competencies [14] and promote progress across all phases of research. Students grow their project in complexity and generalizability over time, as their skills and the research work matures.

- **Doing planning.** In an ARS, students take on the responsibility for planning their work at frequent intervals following agile methodologies. Students record tasks and progress in *sprint logs* that support students' and mentors' awareness of progress and potential needs for replanning.

- **Learning planning.** To help students learn to plan research work on their own, mentors in an ARS provide plan feedback weekly through *special interest group* (SIG) meetings. This meeting facilitates peer review and feedback by mentors and students to help student teams develop their planning skills, devise strategies to overcome challenges, and connect to *resources* [30, 34]. To promote reflection, student teams use *research logs* to record and reflect on research progress throughout a sprint and complete self-assessments in the form of *independent development plans* at quarterly intervals.

ARS address the orchestration challenges of learning to getting help in the following ways:

- **Distributed help.** To better support students while respecting the limits on mentor time, in an ARS the responsibility for providing help is shared across the entire community. Instead of relying on a single mentor to resolve problems, an ARS seeks to make effective use of the diverse sets of expertise that individual community members have by connecting students to those who can best help on a particular

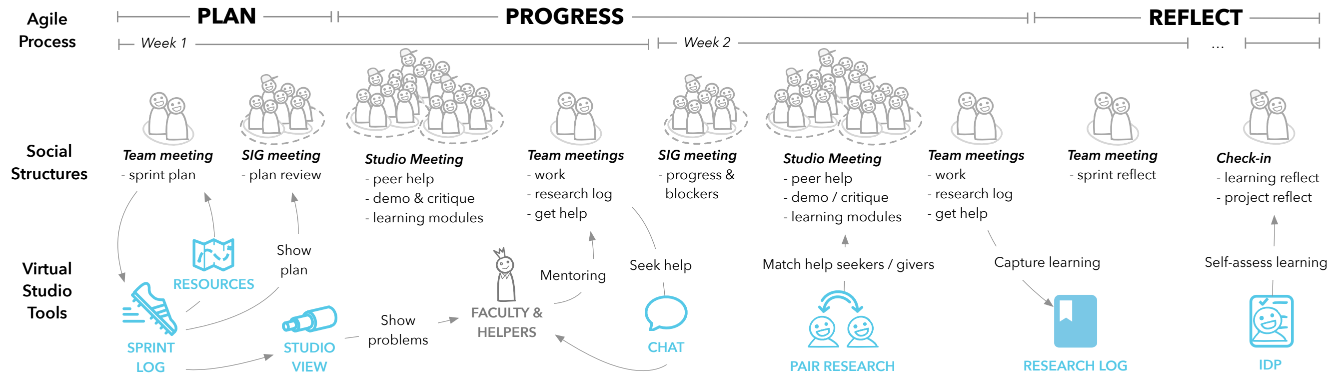

Figure 3. Agile Research Studios support regulation by adapting agile methodologies to research through 2-week sprint cycles of research planning, progress making, and reflection. Social structures help to orchestrate research activities by (a) supporting learning self-directed research planning, monitoring, reflection, and replanning; and (b) facilitating help-seeking and collaboration to promote learning and progress-making. The virtual studio tools extend social structures to more effectively orchestrate learning and support in and out of the classroom.

problem. This should enable the community to fulfill numerous help requests without using mentor time, and lead to students feeling more supported.

- **Scaffolding help-getting.** To help students connect to peers who can help them, ARS scaffolds the process of getting help by using *pair research* [35] to match students to help one another, *SIG & studio meetings* to facilitate students connecting to helpful peers and mentors in and out of their SIG, and *chat* programs such as Slack to enable students to seek and receive help on-demand.
- **Learning help-seeking.** ARS normalizes help-seeking and trains students to seek help effectively. Further, as students are connected to help and help themselves with the support of the above mentioned scaffolds, we expect the common practice of getting and giving help to over time lead to broad shifts in students' help-seeking dispositions [40].

AGILE RESEARCH STUDIOS

Having presented our research questions and design arguments, we provide in this section a detailed description of the ARS model, as it is implemented in the Design, Technology, and Research program. Figure 3 presents an overview of the components and interactions among the processes, social structures, and tools that collectively describe how we propose to create Agile Research Studios.

Agile Research Studios support a research community of undergraduates, graduates, and faculty with varying interests, expertise, and experiences to (learn to) conduct research. A studio is typically organized around a broad research theme that matches a faculty mentor's research interest. Studios can fit within typical academic calendars and allow students to continue for multiple quarters or semesters to promote deep, interest-driven learning and to develop research skills and products over time. *All students, regardless of seniority, conduct independent research and receive authentic research practice.* Students and mentors pursue projects that match their mutual interests. Each project team is kept small (1–3 people) to promote research ownership.

Agile Methodologies

Agile Research Studios support students' development of regulation skills by adapting agile methodologies [46, 13, 31, 29] to research through *sprint* cycles. Each sprint is a set period of time (e.g., 2 weeks) during which students plan specific work to be completed that delivers value to the research project. Student teams meet over the course of each sprint to plan research work, make and share progress, identify difficulties early, replan, and reflect on progress to improve future plans. They also regularly receive coaching and feedback from mentor and peers who help them (learn to) set appropriate and feasible goals, and devise strategies for making progress and overcoming blockers.

Social Structures

To support student researchers learning regulation skills within a community of practice and to scale faculty time, we propose three social structures to orchestrate shared regulation of learning throughout each sprint:

1. **Team meetings** bring together students on the same project to: (a) make sprint plans; (b) perform research; (c) get in-team help; (d) record progress through *research logs* [39]; and (e) reflect on sprint outcomes.

2. **Special Interest Group meetings** (SIG meetings) bring together undergraduate students, graduate students, and faculty working on different projects in the same research area [50]. Each SIG is its own mini-studio initially led by a faculty member whose leadership fades over time as a graduate student *SIG lead* gains competencies in mentoring and becomes the leader of their own SIG. At the start of a sprint, teams share the outcome of their last sprint and present their current sprint plans for review. Halfway through a sprint, teams present their progress and SIG members help devise strategies for overcoming blockers.

   The purpose of planning together is to help students gain planning skills [30]; student receive coaching and feedback from mentors and peers to increase learning [15, 1, 28, 27]. Mentors and peers prompt teams to: (a) describe how their sprint goals connect to their larger research goals; (b) clarify what the actual deliverables of the sprint would be, and what value it would have; (c) consider alternative plans; (d) assess whether tasks can feasibly be completed in time; and (e) generate strategies for better *slicing* [14] work. While newer students receive more scaffolding and direction, senior students are challenged to demonstrate and practice their regulation skills in formulating and revising plans.

3. **Studio meetings** bring together all researchers in a studio to promote progress, learning, and collaboration across SIGs. The studio meeting consists of: (a) *work time*, during which students work on their own projects, (b) *peer help*, during which students formally or informally seek and receive help from others; (c) *demo & critique*, which allows students to share work progress and receive valuable feedback from the entire community; and (d) *learning modules*, which are led by faculty members and students to share "tools of the trade" and teach research skills that benefit all students. Studio meetings embed students in a larger community of practice beyond the problem domain of their SIG. Multifaceted community support provided via multiple mechanisms address diverse students needs at different stages of research and their personal development.

These social structures support meeting in-person to plan work, conduct research, provide and receive help, and reflect and share progress. Depending on its purpose, meetings can be 1-on-1, in small groups, and across the entire studio. To further support student reflection and developing over time, bi-quarterly *check-in* meetings facilitate faculty and graduate student mentors meeting with students individually and in their teams to reflect on their learning and project progress.

Virtual Studio Tools

To effectively orchestrate learning and instruction, we have developed *virtual studio tools* that extend social structures and in-person meetings to orchestrate learning and support progress-making in and outside of the classroom [24]. Virtual studio tools support 3 interrelated feedback loops: (a) a sprint planning and replanning loop in which students and mentors receive feedback on project plans and progress, (b) a help and collaboration loop that helps students scope help requests and connects to helpers, and (c) a reflection loop that promotes awareness of learning and project progress to support growing over time. We describe below the affordances that virtual studio tools should provide:

1. **Sprint Logs** are interfaces that allow student teams to record all tasks they plan to do for a sprint, update task progress throughout a sprint, and replan as needed (see Figure 4). Following professional sprint planning practices, students use the sprint logs to enter high-level deliverables, or *stories*, and the *tasks* for completing those stories. To prioritize work, students assign *points* to stories and tasks to estimate the value of the work to the research and the effort required to complete it. This helps students to think through the process of scoping out work that they can feasibly accomplish within a sprint that deliver value for their research. During a SIG meeting, displaying the

Team	Points Available	Points Committed	D	T	R	Hours Spent	D	T	R	Progress
[illegible]	35	35	12	8	15	19.75	5	7	8	56%
[illegible]	16	19	1	17	2	6	1	6	0	32%
Total	**51**	**54**	**13**	**25**	**17**	**25.75**	**5.75**	**13**	**8**	**48%**

Stories	Tasks for Story	Points Required	D	T	R	Assigned To	Status	Hours Spent	Helpful Links
Have a functional tracking protoype that can track a runner's location and prepare data to be sent to a cheerer	*start entering tasks for this story on the next line ↓*	**17**	*mark*	*mark*	*mark*	*enter your name below to pick up tasks ↓*	*mark as: in progress, backlogged, or done*		
	pseudocode tracking protocol & structs	1		x		Leesha	done		pseudocode doc
	read Swift guide for protocols/syntax	2		x		Leesha	done	2	swift protocol docs
	go through Ray Wenderlich tutorial on POP	2		x		Leesha	backlogged		protocol oriented programming
	implement tracking protocol & structs	3		x		Leesha	in progress	5	
	implement tracking protocol & structs	5		x		Christina			
	Test tracking for cheerer	0.5		x		Christina			
	test tracking for runner	0.5		x		Christina			

Figure 4. Screenshot of a spreadsheet prototype of a project team's *sprint log*. The top half of the sprint log provides an overview of commitments, hours spent, and progress on the current sprint. Students plan their sprints in the bottom half by recording high-level deliverables, or *stories*, and the *tasks* for accomplishing those stories. Students use a *point system* to estimate required effort to avoid committing more time than they have available for the sprint. As students make progress they mark tasks as *done*, *backlogged*, or *in progress* and record hours spent. Students also link to useful resources next to stories and tasks.

sprint log facilitates students, peers, and mentors communicating their plan, reporting progress, and devising strategies to overcome blockers.

2. **Resources** are references and guides that describe how to achieve commonly shared stories and tasks, such as writing a paper or setting up a technology. While students work on different projects, within an ARS there is likely to be significant overlap in methodology that allow for similar resources and guides to be useful across projects. We curate these resources and provide affordances for students to find helpful resources from their sprint log and to suggest resources to others during SIG meetings.

3. **Studio Views** are dashboards that summarize the sprint logs of all teams. This helps surface potential problems to mentors by helping them see at a glance progress across teams. For example, a mentor can see which teams are behind and connect with those teams for a quick check-in or an impromptu office hours to resolve larger challenges.

4. **Chat** is a collection of online chatrooms, each with a dedicated topic. Chat provides a medium for students to reach out to mentors and peers for help online and on-demand. We replicate the social structures of Agile Research Studios by providing chat rooms for project teams, SIGs, and the whole studio. We also create rooms to support other community needs such as planning fun activities or discussing interview candidates. Team and studio members can connect through private messaging, and use bots to stay aware of others' activities and facilitate progress updates.

5. **Pair Research** [35] is a system for pairing students to help one another on their respective projects based on their reported task needs and ratings of how well they can help others. By finding globally optimal matches across a studio, pair research distributes help across the community of learners within studio meetings to make effective use of the community's collective expertise and time resources. This in turn frees up faculty and graduate student mentors to provide help where they are most needed.

6. **Research Logs** are personal diaries in which students are prompted to capture and reflect on their research progress and learning [42]. The research log serves primarily as a research diary rather than a tool for communicating plans.

7. **Independent Development Plans (IDP)** are a set of close- and open-ended questions about project progress, including what challenges students encountered, how they overcame challenges, what challenges remain, and perception of personal growth. IDPs help students self-assess their learning and progress. Questions are divided into sections on Research Work, Collaboration, Growth, Research Process, and ARS Process to promote students reflecting on different facets of their development and work progress.

We have created initial prototypes of the virtual studio tools described above by using, building on, and integrating existing and free collaboration and communication tools including collaborative editors and team messaging programs. Figure 4 shows a Sprint Log we prototyped using Google Spreadsheets that supports student teams planning their work, sharing their work progress, and linking to helpful resources in Google Drive and from the Web. A spreadsheet-based Studio View prototype loads student teams' sprint log data and displays to mentors how far along teams were on their sprint versus how much time was left in the sprint. We use Slack as our chat program and created a spreadsheet prototype for Pair Research. Students maintain research logs using Google Docs and store their project files in personal cubbies in Google Drive folders. The IDP is implemented as a Google Spreadsheet.

Participation, Project Selection, and Social Norms

With more demand than available mentoring resources, we use structured interviews following industry established best practices [7] to select students who are prepared to make research contributions within a research learning community. Similar to advanced classes and labs, students joining an ARS are required to have the requisite subject area expertise. Beyond subject area expertise, we select for students who show aptitude for embracing collaboration, growing through challenges and failures, taking initiative, managing ambiguity, and seeking deep understanding [49]. Given limited interviewing resources, students in an ARS interview candidates together with the faculty mentor and recommend other students who they think are good candidates for joining the ARS.

To promote student engagement and ownership of their project, students select the project they work on in the SIGs that they are interested in. The faculty mentor meets with incoming students to brainstorm project ideas and discuss project fit and the technical training necessary. The mentor helps students identify a set of scoped project ideas that are feasible for the student and are likely to lead to research contributions of interest to the SIGs the student is interested in. Students have the final say on which project they choose from this set. Students may change projects when their project comes to a close or their interests change; the faculty mentor also supports students selecting a new project in such cases.

An ARS may establish a variety of social norms to promote a culture of collaboration and to help students build regulation skills. For example, to promote effective help-seeking, students may be frequently reminded by mentors to ask for help if they are stuck for more than 30 minutes. To emphasize the importance of learning regulation skills, students may receive positive feedback from mentors for trying to regulate (e.g., exhibit understanding of their planning processes, or trying to clarify goals and decide among alternative plans) rather than praise for students' successful work outcomes. This is consistent with helping students develop a growth mindset [20] for learning and practicing regulation skills.

METHOD

We take a design-based research (DBR) approach to develop the ARS model for overcoming the orchestration challenges of research training. DBR [11, 8, 12, 38, 22] requires iterative cycles of defining design arguments (hypotheses), implementing the design argument, collecting data, evaluating the design, and refining the design. We study the ARS model through a case study of the Design, Technology, Research (DTR) program, a research learning environment of 20+ students implemented and led by one of the authors.

Context: Design, Technology, and Research (DTR)

DTR was implemented at a private US research university in Spring 2014. The goal of DTR is to realize and develop undergraduate and graduate students' potential for developing novel technologies and creative solutions through design, engineering, and research. Such practice is often lacking in lecture-based classrooms yet is crucial for preparing students for the complex social and technical problems they will face in STEM careers [26]. As participants in this for-credit program, students lead research projects in social and crowd computing, cyberlearning, and human computer interaction.

In repeated 10 week-long studio sessions, students work with mentors to identify a research project, explore and iterate over designs, prototype at varying fidelities, build working systems, conduct evaluative studies, and report findings through progress reports, blog posts, workshops, and conference publications. Students follow the entire research process each session; projects grow in complexity and generalizability over multiple studio sessions.

Students learn about DTR through word of mouth, recruitment emails, and course catalogs. They apply to join the DTR program and receive course credit for enrollment. Student can enroll repeatedly to satisfy computer science major course requirements or as an independent project course. Students are not obligated to stay in the program though the majority of undergraduate DTR students continue until they graduate.

Data Collection and Analysis

To study the participation, productivity, and learning outcomes from the ARS model we collected records of students' enrollment in DTR, tallied major products produced, and analyzed student quarterly self-assessments. We deductively coded the self-assessment reports for students' perceived development of regulation skills (i.e. in planning and help-seeking) and shifts in self-efficacy, attitudes, and dispositions.

Common Obstacles	Student Before	ARS Solution	Student After
Doing all research steps Students aren't involved in all phases of authentic research project (e.g., perform 1 technical piece)	Less experienced students relegated to less central research activities	Mentors use *agile methods* to slice project work to fit student competencies	More students engage in all phases of research activities
Doing planning Students have limited planning skill, so mentors responsible for planning or tolerate projects going off track w/ too many students	Mentor plans projects, students engage in few explicit planning activities	Use agile planning methods and tools (e.g., *sprint logs, resources)* to scaffold planning	Students explicitly responsible for planning activities
Learning planning Scarce mentor time are not used to teach effective planning, but rather used for resolving technical challenges.	Students may develop subject area expertise but don't learn how to plan research work on their own	Mentors provide plan feedback in *SIG meetings*; *research logs* & *IDPs* promote regular reflection	Students develop regulation skills to plan research work on their own

Figure 5. ARS addresses the orchestration challenges for learning to plan research by adapting agile processes to research training so students are responsible for planning research inquiry and learn to plan more effectively over time.

To study how students planned and replanned their work, we used the Google Drive API to collect the revision histories on each project's sprint log over the two completed quarters during which the tool was deployed (Fall 2015 & Winter 2016). Each revision represents a set of edits grouped into a short time period. We also recorded the final status of each sprint, including points commited, hours sprint, tasks completed or backlogged, and resources linked.

To study students' helping behaviors, we surveyed students for the names of people they helped and were helped by in DTR each quarter, as well as what they helped with. This survey was included as part of the self-assessments in the last four quarters (Winter 2015, Spring 2015, Fall 2015, Winter 2016). We coded each instance of help by the type(s) of help requested and fulfilled, divided into subcategories within design, technology, and research. To understand how helping behaviors may be affected by social structures in DTR, in our analysis we combined the helping data with a graph of the closest relationship between each pair of students as teammates, SIG-mates, or studio-mates each quarter.

To understand how students connected online, we collected Slack usage statistics and public message histories during the six-month period from November 7th, 2015 to May 5th, 2016. While Slack's team statistics webpage only surfaces basic stats on the front-end, it contains within its source code a JSON data object with a more detailed breakdown of the number of messages sent in the channels and groups that the authors are a part of. We used this data for our analysis.

DTR PILOT STUDY RESULTS

Our pilot study results provide early indications of the potential effectiveness of the ARS model for providing authentic research training to increased numbers of students.

Conducting and planning research

Figure 5 summarizes how ARS proposes to address the orchestration challenges for learning to plan research. We present results on each of the points below.

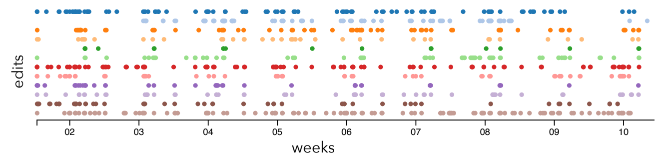

Figure 6. Visualization of revisions to each team's sprint log across the weeks of the Winter 2016 DTR studio. Each row of colored dots presents the edits made by a team. Students planned a shortened 'sprint 0' in the first week; each subsequent 2-week sprint started and ended with SIG meetings scheduled 2-3 days after the start of each week.

		-1D	SIG	+1D	Week
P(≥1 revision)	W16	88%	62%	35%	75%
	F15	72%	57%	24%	54%
# of revisions	W16	2.9	0.7	0.6	3.1
	F15	1.6	0.7	0.4	1.7

Figure 7. The likelihood and average number of sprint log revisions made by teams each week in the day before their SIG meeting (*-1D*), during their SIG meeting (*SIG*), the day following their SIG meeting (*+1D*), and throughout the rest of the week (*Week*).

Doing all research steps

Our pilot of DTR used the ARS model to provide authentic research training to a large number of students, produce multiple research outcomes, and sustain participation over time. Over the last two years from Spring 2014 to Winter 2016, we hosted six academic-year, quarter-long studios with 4 graduate students (3M, 1F) and 32 undergraduate students (22M, 10F). In the Winter 2016 studio, we hosted 21 students (2 PhD, 1 post-bac, and 18 undergraduate) who led 13 research projects.

DTR students iteratively designed, built, and user tested 18 new systems. 19 students have received university undergraduate research grants. 8 student-led full papers and extended abstracts have been accepted at ACM and AAAI conferences. Further, three students placed 1st, 2nd, and 3rd at ACM CHI and Grace Hopper student research competitions.

While students were not obligated to stay in DTR beyond a quarter, 94% of students completed at least two quarters of DTR (34 out of 36 students). Students who have graduated stayed in DTR for an average of 3 quarters; 64% of them continued in DTR until they had graduated (14 out of 22 students). 12 out of the 36 students left DTR before graduating. They cited a number of reasons, including being more interested in building technology than the research work (4), needing to finish course requirements to graduate on-time (4), switching to another research area and lab (2), and having completed a PhD rotation (2).

Doing Planning

DTR students explicitly engaged in planning activities by updating their sprint logs regularly. In Fall 2015 when sprint logs were first introduced, student teams made an average of 4.4 revisions per week (515 revisions total). By Winter 2016, students made an average of 7.3 revisions per week (857 revisions total). Figure 6 shows that teams in Winter 2016 made edits to their sprint log throughout the week, and almost all teams made at least one edit each week.

Looking more closely at when students made edits, Figure 7 shows that while students made a majority of their edits throughout the week and in particular in the day prior to the SIG meeting, they also regularly made edits during the SIG meeting and immediately after (62% and 35% of the time in Winter 2016, respectively). This is consistent with our observations of students updating their sprint and adding resources based on feedback and suggestions from peers and mentors during SIG meetings, and of student teams meeting up after SIG meetings (sometimes immediately after) to sync up and replan their sprints to refocus immediate project goals.

Students were fairly accurate in estimating how long it would take to complete their sprints, but did have a general tendency to underestimate. Students' committed points were within 20% of students' reported work times on 70% of sprints (59 out of 84 sprints); students took even longer to finish their commitments 20% of the time (17 out of 84 sprints). To compensate for some stories taking longer than anticipated, students backlogged 23% of their tasks (468 out of 2,003 tasks) for completing in a future sprint.

Students followed a number of other planning practices recommended by faculty and graduate student mentors. Students were advised to stay within the points allotted for each sprint; in only 3 out of 84 sprints did students spend more hours than the recommended point allotment by more than 20%. Students were encouraged to identify resources that may help them complete stories on their sprint; students recorded 120 resources as 'helpful links' that they found or were suggested by peers and mentors (5.5 per team per quarter).

Learning Planning

From self-assessments, students reported developing planning skills to drive effective research inquiry by using sprint planning to help them break down big tasks into smaller goals, prioritize goals, and to "see what success looked like at every step." Students reported developing a number of strategies and skills for delivering value within two week sprint cycles, including (a) building at the fidelity appropriate for the current stage of research, (b) prioritizing important features and research questions, (c) sequencing tasks, (d) defining concrete outcomes, and (e) moving on despite uncertainty or imperfect knowledge. Students noted learning the importance of being able to reason about alternatives and to understand tradeoffs. Beyond individual sprints, students also reported learning to perform "careful prior planning with end goals in mind" at the quarter level to ensure that they completed studies and met paper deadlines.

Many students struggled initially with effectively planning their design and technical work around their research contributions. Instead of focusing on stories that yielded clear research value (e.g., that tests a design argument), students planned and spent too much time on stories around complex features that were less critical to the research. With mentoring, these failures led students to develop more effective strategies over time. Students reported learning to conduct iterative, small-scale tests to orient their research direction, prioritizing tech features to answer research questions, syncing their hypotheses and study designs to their technology, and avoiding "getting sidetracked by other nifty design features and remember that everything has to be tied back to research outcomes."

Common Obstacles	Student Before	ARS Solution	Student After
Distributed help Students need diverse technical help to make progress. Busy mentors don't have time to help.	Students wait for mentors and do not always feel supported	GIve the entire community the responsibility for providing help	All students are helping; students feel supported by community
Scaffolding help-getting Even when distributed help is available, students do not always get help due to friction (e.g., not knowing who can help or finding it tedious to connect to help)	Students don't receive the help they need to make progress	Connect students to peers using *pair research*, *chat*, *studio views*, and *SIG & studio meetings*	Students regularly get diverse help from community members
Learning help-seeking Students are reluctant to ask for help and lack help-seeking skills and dispositions	Students are reluctant to ask for help from mentors and peers in the community	Help-seeking is normalized and trained; scaffolds and practices connect students directly to help	Students develop help-seeking skills and dispositions

Figure 8. ARS addresses the orchestration challenges for learning to get help by supporting students (learning to) make effective use of the expertise and resources across the research learning community.

	N	Total	Median	Average	% of studio
Winter 2016	21	160	7	7.6	38%
Fall 2015	14	61	4	4.4	34%
Spring 2015	14	65	5	4.6	36%
Winter 2015	15	86	5	5.7	41%
Total/Average	33	372	5	5.8	37%

Figure 9. Helping statistics showing the number of fulfilled help requests reported in different quarters of DTR, the average/median number of people receiving help, and the % of people in the studio receiving help.

Students noted that SIG meetings and follow up discussions with faculty mentors and SIG leads helped them learn to better plan at appropriate fidelities, see the big picture and refocus on higher-level research goals, refine research directions and questions, and to better manage team issues. Students also report that talking to faculty mentors and SIG leads helped them to set appropriate expectations, and to reorient their perspective on research given frustrations and problems. As developing mentors, SIG leads expressed that they felt that they were able to help students drive their research, and expressed a desire to continue learning to more effectively manage young researchers.

Getting help and making progress

Figure 8 summarizes how ARS proposes to address the orchestration challenges for learning to get help. We present results on each of the points below.

Distributed help

DTR students helped over a third of the other students in their studio in any given quarter, and fulfilled 372 help requests over the four quarters in our dataset. See Figure 9 for the helping statistics from each quarter. DTR students were willing to help others working on different projects in their SIG and across the studio; 329 reported help requests (89%) were fulfilled by a student working on a different project. 115 requests were fulfilled by a fellow SIG member working on a

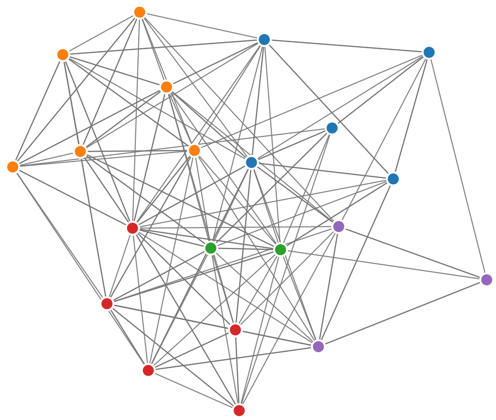

Figure 10. Helping graph from Winter 2016. Each node represents a student and links represent fulfilled help requests. Students helped and received help from both students in their own SIG (same color node) and across the studio.

different project (31%), and 214 requests were fulfilled by a student in another SIG (58%). Figure 10 shows a dense network of helping behaviors from the Winter 2016 that connects students both within and across SIGs.

Students helped other students who also helped them in return. Of the 329 help requests made by students working on different projects, 196 requests were reciprocated in the same quarter (60%) and 231 requests were reciprocated across quarters (70%). Students also helped other students when they did not receive help in return. Many students expressed wanting to pay it forward, and felt that it was natural to want to help others given all the help they had received in the past. One student said "Nobody feels guilty receiving help because they know they've helped others as well."

Students reported that seeing other people in DTR's "love for helping others" led them to want to cultivate and spread the helping culture by both "share [their] experience and utilize others." Students who initially dismissed their own abilities recognized that "there's always something [they] can help someone else with." Students also commented that they enjoyed giving help to others who needed it; they noted that helping others allowed them to "get a much better sense of what everyone else was doing" and usually gave them additional ideas that benefitted their own project and learning.

Scaffolding help-getting

Students received help from community members on a wide variety of requests across design (n = 159), technology (n = 129), and research (n = 118). Figure 11 shows help requests across these areas by category. Popular requests include asking others to test the latest version of a prototype, helping with web/mobile development (e.g., building a Chrome extension or deploying an iOS app); and refining research directions. Help across requests for design, technology, and research were fulfilled by team, SIG, and studio members at about the same proportions. Among major subcategories, there were two instances where significantly more help was provided by one group than another. This includes help on research directions, where 34 of the 41 requests were fulfilled

Design help (n=159)		Technology help (n=129)		Research help (n=118)	
PROTOTYPE TESTING	103	WEB/MOBILE DEV	81	RESEARCH DIRECTION	41
NEEDFINDING	45	SYSTEM FEATURES	20	STUDY DESIGN	33
BRAINSTORMING	15	DEBUGGING	17	PAPER WRITING	32
FEEDBACK	16	PAIR PROGRAMMING	14	DATA ANALYSIS	15
INTERACTION DESIGN	16	ARCHITECTURE	12	GRANTS	14
UI/UX DESIGN	9	GENERAL	10	USER STUDIES	10
RECRUITING	8	DEV TOOLS	7	RELATED WORK	9
PROTOTYPE DESIGN	4	ALGORITHMS	3		

Figure 11. Fulfilled help requests by category.

```
Channels     8403 (100%)
  Team       4948 (59%)    project discussion among teammates (and w/ mentors)
  Studio     1131 (13%)    studio-wide announcements and water cooler discussion
  Bot        1063 (13%)    GIT (code) and Youtube integration (sprint videos)
  Community   416 (5%)     swag, senior gifts, 20% time, job search, fun activities
  SIG         305 (4%)     sharing resources and communicating within a SIG
  Faculty     240 (3%)     faculty discussion about agile research studios
  Interview   211 (3%)     committee coordinating and scheduling DTR interviews
  SIG Heads    71 (1%)     SIG head and faculty check-in
  Other        23 (<1%)    occasional use; testing bots
```

Figure 12. Slack channels and message counts by group and purpose.

within a SIG (84%), and for prototype testing, where 72 out of 103 requests were fulfilled by studio members across SIGs (70%). These numbers are consistent with ARS structuring SIGs around particular areas of research focus, and show students reaching more testers by recruiting across the studio.

Students reported that studio meetings helped them learn from other students in the studio and broadened their perspective on their own projects and on research more generally. Students noted that demo-critique sessions helped them brainstorm new ideas, clarify research directions, and gain fresh perspectives. Students benefitted from pair research and found it to be a "fantastic way to scale a class where a professor doesn't have time/specific skills to give each student that much individual attention, and when students can help each other." Students also found that the learning modules led by SIG leads and faculty mentors helped stretch their thinking, for example to consider validity concerns when designing a study, architecting apps that can scale to millions of users, and understanding realistic research timelines and the need to plan for failure.

Outside of in-person meetings, DTR members communicated extensively with one another via Slack. Over a six-month period, DTR members sent a total of 36,919 messages and shared 660 files. Of these messages, 8,407 of them (23%) are in visible public and private channels the authors are a part of; the rest of the messages are direct messages or other private group conversations. Figure 12 shows the number of messages sent through visible channels categorized by a diverse set of purposes, and the members who participate in them.

Learning help-seeking

Students overwhelmingly reported an increase in their willingness to seek out help as they participated in DTR. Students who were initially reluctant to ask for help learned that "[they] can ask for help and that everyone asks for help and it doesn't make them stupid to need help." One student noted that she reached out more easily once she realized that "people in DTR could help me out much more than I could get by myself." Students reported adopting a number of effective help-seeking strategies rarely practiced by undergraduate students [36, 37, 41], including doing a little bit on their own to help them formulate help requests better, asking for help online and offline, “expanding [their] network of help-givers and receivers with the new class [of students],” and learning to become more comfortable asking for help by starting now.

Students recognized the importance of recognizing blockers that impeded their progress and seeking help to overcome them. Through less productive sprints, students learned that it “is detrimental to try to work through blockers on your own. Asking for help should be the first step when you really get stuck on a blocker.” Students noted their struggles in “admitting that something I’m doing is not working” and being “adamant in fixing things on my own.” They commented that DTR “taught me to acknowledge when I need help and that it’s perfectly acceptable and important to ask for that help.” Students also acknowledged the need to ask for help quickly, to not wait till the next SIG meeting and to instead reach out to mentors “more often as that could have prevented the issues I faced or at least I would have discovered earlier.” These realizations are rare among undergraduate students outside of DTR yet critical for supporting student learning and progress-making in project-based learning environments [23].

Orchestration Time

With the community structure for planning and support in place, the faculty mentor was able to orchestrate DTR with 2 PhD, 1 post-bac, and 18 undergraduate student researchers leading 13 research projects with less than 12 hours of faculty time each week. The faculty member used: 5 hours for SIG meetings (five SIGs each with up to 3 active projects and 6 students), 3 hours for the all studio meeting, and 4 hours for in-person and virtual help to respond to students on demand. On typical weeks, the faculty mentor was able to maintain awareness of progress across projects and had time to respond to research challenges that they themselves can best address. On “surge weeks,” e.g., when approaching paper and grant deadlines, the faculty mentor spent considerably more time reviewing drafts, editing, and helping out in any ways needed.

DISCUSSION

In summary, our pilot study of DTR showed that the practices, structures, and technologies in an Agile Research Studio empowered undergraduate students to plan research work at weekly intervals and overcome challenges quickly with the support of peers and mentors. This allowed them to conduct independent 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.

We discuss below how the ARS model can impact research training beyond DTR, the limitations of our study, and future work in learning and orchestration technologies.

Implications for Research Training

DTR pilot results suggest that ARS scales mentor time by distributing the responsibility of research planning and connecting students to teammates, SIG members, and the rest of the studio for help. This frees up faculty mentors to focus on teaching regulation skills and readily responding to challenges they can best address. As students develop regulation skills, they become less reliant on the mentor; this further scales mentors’ time and allows them to more skillfully direct efforts based on research importance and student needs.

DTR pilot results suggest three ways in which ARS supports students produce research outcomes. First, by engaging students in planning and providing plan feedback, students learn to deliver research value with each sprint, prioritize research goals, and avoid spending time on less crucial tasks. They also (learn to) catch problems earlier and flexibly replan with the support of mentors and peers. Second, by promoting helping behaviors, students are able to receive the help they need to overcome blockers and make regular progress. Third, by scaling mentor time, ARS significantly expands the number of student-led project producing research outcomes.

Students in an ARS are responsible for not only their own learning and progress, but that of other members of their studio. Establishing a supportive community in which students take on the responsibility of helping one another is crucial for an ARS to exhibit the outcomes observed in DTR. By using practices such as pair research—where direct reciprocity is built in—but also scaffolding and encouraging help-seeking and help-giving more generally, ARS involves growing a supportive community over time in which generalized reciprocity is commonly practiced [4].

While the ARS model allowed DTR to scale a research learning community with over 20 members, it may also benefit faculty members who run typically-sized research labs (e.g., 5–10 students). Training students through the ARS may still be worthwhile for advancing research productivity over time as long as the mentor’s investment in developing regulation skills pays. Despite having fewer helpers with possibly less diverse expertise, ARS support for help-seeking and help-giving still scales mentor time in ways that promote learning and productivity. While the benefits may be somewhat smaller for smaller labs, in these ways ARS can still provide significant benefits and also pave way to lower participation barriers to include undergraduate students and train more students.

The ARS model may also be useful for supporting a community much larger than DTR by further distributing faculty responsibilities. One challenge is ensuring that there is enough mentoring resources to support regulation skill development. Communities can scale up in size as graduate students’ develop their mentoring ability and become more ready to lead their own SIG; this allows faculty mentors to fade from more established SIGs to start new ones. Another challenge is connecting students to help across a larger community. This imposes additional orchestration burdens, but also challenges in establishing a helping culture should students feel less connected. New orchestration technologies and effective community designs thus become increasingly important as the community expands in size.

ARS provides a powerful example of how socio-technical systems can be designed to support cooperative work in ways that allow us to provide research experiences to more students. Support for mentoring self-directed learners, leveraging distributed expertise, and promoting awareness of needs reduce orchestration burdens to scale mentor time and advance work outcomes. We believe socio-technical solutions for supporting cooperative work will play an increasingly important role in empowering us to provide authentic learning experiences to many more students to prepare them for tackling complex challenges in the 21st century [49].

Study Limitations

Our exploratory study has limitations including: measurements of learning, author as researcher and designer, participant selection bias, and isolation of factors.

Measurements of Learning

While we used student self-assessments to surface students' perceptions of their learning and growth, we did not directly measure learning and growth. Self- assessments may be biased as students may under- or overstate their learning and skills. To complement these initial measures, we also collected and analyzed traces of student interactions with one another and with the virtual studio tools; this provided evidence that students were following regulatory processes and being mentored to improve their regulatory processes. Prior research shows that the practice and mentoring of regulation skills helped students develop them in a number of science domains [1]; future work can attempt to measure such gains directly in DTR. Consistent with design-based research best practices [3, 22], we use these measures to quickly identify failures in the design and to iterate, with plans to directly measure learning as the design show promise.

Author as Researcher and Designer

Design-based research allows researchers the control to simultaneously iterate on and study complex models [3, 22]. However, this introduces the possibility of bias as one of the researchers was also the faculty member in the design. To limit the risks of biasing results, we centered our analysis primarily on student self-reports and log data. While student self-reports may still be biased in favor of the desired learning outcomes, students were also forthcoming about their struggles with regulation. Beyond biasing results, designing for our own lab runs the risk of creating solutions that don't work well in other settings. To mitigate these risks we have focused on common orchestration challenges informed by the learner literature and designed tools that are largely domain-agnostic (e.g., pair research, sprint logs). But even so, some adoption challenges remain. As one example, while agile methodologies work naturally with the design- and technology-centered research work in DTR, adopting such methods to other fields of study may require new ways of working that are less familiar. In future work we are interested in supporting other research communities adopting the ARS model and studying its effectiveness and any adoption challenges across sites.

Participant Selection Bias

This paper argues that ARS can significantly scale mentor time and increase research productivity all while training and relying on undergraduate researchers. However, we do not make a strong claim that this will work for a randomly selected population of undergraduates as the ARS model specifically includes screening undergraduates who have the technical skills and career interests required to conduct undergraduate research. This raises a concern about the following counterfactual: perhaps students selected for DTR are more qualified than those who typically engage in research, and that their learning and research accomplishments are attributed solely to selection. However, anecdotal comparisons suggest this to be unlikely. First, traditional 1-1 apprenticeship models rely on an even more stringent selection criteria, selecting the best applicants for graduate school and then selecting admitted graduates based on their fit for the lab. ARS lowers the floor for research by widening the pool to qualified students to include many undergraduate sophomores. Second, comparing to an honors thesis course at the same university, which has a similar selection requirement and can involve upwards of 30 hours per week, students publish infrequently if at all. Of course, these comparisons are only anecdotal–future work must more rigorously measure how selection effects influence ARS outcomes relative to other programs.

Isolation of Factors

While our analyses provide evidence that the ARS model led to the observed outcomes, one study is not sufficient to isolate all the factors necessary to DTR's efficacy. As one example, while we observed that DTR students were eager to help others, we cannot say at this point that adopting the ARS model as described is sufficient for creating a strong community culture in which students are as willing to help as DTR students were. We look forward to refining the ARS model over time as we continue to develop our understanding of other design considerations that may be critical to a studio's success.

Future Work in Learning and Orchestration Technologies

Advancing socio-technical platforms for orchestrating research training can help to (a) scale mentor time, (b) produce effective research, and (c) engage more students to conduct independent research. Our pilot results suggest that the virtual studio tools we have prototyped are already helping to support these goals by engaging students in planning and helping activities, surfacing needs for help-seeking and replanning, and connecting students to help and instruction. A core focus in future work is to advance socio-technical platforms that orchestrate learning and instruction outside of in-person meetings to support students learning regulation skills and becoming aware of needs for mentoring and support. For planning, learners still lack scaffolds for planning effectively on their own and can struggle to reprioritize tasks to deliver research value and fail to recognize their needs for help. For help-seeking, while students use Slack extensively outside of in-person meetings, they still struggle with formulating help-requests, especially for more open-ended help on topics such as framing design arguments or refining research directions. Related, students can benefit from tools that provide additional scaffolds for connecting to available mentors and peers on-demand. Advancing project-based learning platforms capable of monitoring learning activities and progress across the studio and triggering and scaffolding help and instruction can play a significant role in advancing research training at scale.

REFERENCES

1. Susan A Ambrose, Michael W Bridges, Michele DiPietro, Marsha C Lovett, and Marie K Norman. 2010. *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons.
2. Alan Bain and Mark Weston. 2012. *The learning edge: What technology can do to educate all children*. Teachers College Press.
3. Brenda Bannan-Ritland. 2003. The role of design in research: The integrative learning design framework. *Educational Researcher* 32, 1 (2003), 21–24.
4. Zygmunt Bauman. 1993. Postmodern ethics. (1993).
5. Molly Beisler and Ann Medaille. 2016. How Do Students Get Help with Research Assignments? Using Drawings to Understand Students' Help Seeking Behavior. *The Journal of Academic Librarianship* (2016).
6. Katerine Bielaczyc and Allan Collins. 1999. Learning communities in classrooms: A reconceptualization of educational practice. *Instructional-design theories and models: A new paradigm of instructional theory* 2 (1999), 269–292.
7. Laszlo Bock. 2015. *Work rules! Insights from Inside Google That Will Transform How You Live and Lead*. Twelve.
8. Ann L Brown. 1992. Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *The journal of the learning sciences* 2, 2 (1992), 141–178.
9. Ann L Brown, Doris Ash, Martha Rutherford, Kathryn Nakagawa, Ann Gordon, and Joseph C Campione. 1993. Distributed expertise in the classroom. *Distributed cognitions: Psychological and educational considerations* (1993), 188–228.
10. Ann L Brown and Joseph C Campione. 1996. *Psychological theory and the design of innovative learning environments: On procedures, principles, and systems.* Lawrence Erlbaum Associates, Inc.
11. Jody Clarke, Chris Dede, D Jass Ketelhut, Brian Nelson, and C Bowman. 2006. A design-based research strategy to promote scalability for educational innovations. *Educational Technology* 46, 3 (2006), 27.
12. Paul Cobb, Jere Confrey, Andrea diSessa, Richard Lehrer, and Leona Schauble. 2003. Design experiments in educational research. *Educational researcher* 32, 1 (2003), 9–13.
13. Alistair Cockburn. 2006. *Agile software development: the cooperative game*. Pearson Education.
14. Mike Cohn. 2004. *User stories applied: For agile software development*. Addison-Wesley Professional.
15. Allan Collins and Manu Kapur. 2005. Cognitive apprenticeship. In *The Cambridge handbook of the learning sciences*, R Keith Sawyer (Ed.). Cambridge University Press, Chapter 6.
16. Allan Collins, Manu Kapur, and R. Keith Sawyer. 2014. *Cognitive Apprenticeship* (2nd ed.). Cambridge University Press, New York, 109–127.
17. Pierre Dillenbourg and Patrick Jermann. 2010. Technology for classroom orchestration. In *New science of learning*. Springer, 525–552.
18. Erin L Dolan and Deborah Johnson. 2010. The undergraduate–postgraduate–faculty triad: unique functions and tensions associated with undergraduate research experiences at research universities. *CBE-Life Sciences Education* 9, 4 (2010), 543–553.
19. Joanna C Dunlap. 2005. Problem-based learning and self-efficacy: How a capstone course prepares students for a profession. *Educational Technology Research and Development* 53, 1 (2005), 65–83.
20. Carol Dweck. 2006. *Mindset: The new psychology of success*. Random House.
21. Carol S Dweck. 2000. *Self-theories: Their role in motivation, personality, and development*. Psychology Press.
22. M Easterday, D Rees Lewis, and E Gerber. 2014. Design-Based Research Process: Problems, Phases, and Applications. In *Proc. of International Conference of Learning Sciences*, Vol. 14.
23. Peggy A Ertmer and Krista D Glazewski. 2015. Essentials for PBL implementation: Fostering collaboration, transforming roles, and scaffolding learning. *Essential Readings in Problem-Based Learning* (2015), 89.
24. Thomas A Finholt and Gary M Olson. 1997. From laboratories to collaboratories: A new organizational form for scientific collaboration. *Psychological Science* 8, 1 (1997), 28–36.
25. Mary Frank Fox. 1992. Research, teaching, and publication productivity: Mutuality versus competition in academia. *Sociology of education* (1992), 293–305.
26. Elizabeth M Gerber, Jeanne Marie Olson, and Rebecca LD Komarek. 2012. Extracurricular design-based learning: Preparing students for careers in innovation. *International Journal of Engineering Education* 28, 2 (2012), 317.
27. John Hattie. 2009. The black box of tertiary assessment: An impending revolution. *Tertiary assessment & higher education student outcomes: Policy, practice & research* (2009), 259–275.
28. John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research* 77, 1 (2007), 81–112.
29. Michael Hicks and Jeffrey S Foster. 2010. Score: Agile research group management. *Commun. ACM* 53, 10 (2010), 30–31.

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

31. Jake Knapp, John Zeratsky, and Brad Kowitz. 2016. *Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days*. Simon & Schuster.

32. David Lopatto. 2007. Undergraduate research experiences support science career decisions and active learning. *CBE-Life Sciences Education* 6, 4 (2007), 297–306.

33. Carol A Lundberg and Laurie A Schreiner. 2004. Quality and frequency of faculty-student interaction as predictors of learning: An analysis by student race/ethnicity. *Journal of College Student Development* 45, 5 (2004), 549–565.

34. Mariel Miller and Allyson Hadwin. 2015. Scripting and awareness tools for regulating collaborative learning: Changing the landscape of support in CSCL. *Computers in Human Behavior* (2015).

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

36. Sharon Nelson-Le Gall. 1981. Help-seeking: An understudied problem-solving skill in children. *Developmental Review* 1, 3 (1981), 224–246.

37. Paul R Pintrich. 2004. A conceptual framework for assessing motivation and self-regulated learning in college students. *Educational psychology review* 16, 4 (2004), 385–407.

38. Tjeerd Plomp. 2009. Educational design research: An introduction. *An introduction to educational design research* (2009), 9–35.

39. Sadhana Puntambekar and Janet Kolodner. 1998. The Design Diary: Development of a Tool to Support Students Learning Science By Design. In *International Conference of the Learning Sciences '98*. ICLS, 230–236.

40. Allison M Ryan and Paul R Pintrich. 1997. " Should I ask for help?" The role of motivation and attitudes in adolescents' help seeking in math class. *Journal of educational psychology* 89, 2 (1997), 329.

41. Allison M Ryan, Paul R Pintrich, and Carol Midgley. 2001. Avoiding seeking help in the classroom: Who and why? *Educational Psychology Review* 13, 2 (2001), 93–114.

42. Donald Schön. 1987. *Educating the reflective practitioner*. Jossey-Bass.

43. P Wesley Schultz, Paul R Hernandez, Anna Woodcock, Mica Estrada, Randie C Chance, Maria Aguilar, and Richard T Serpe. 2011. Patching the Pipeline Reducing Educational Disparities in the Sciences Through Minority Training Programs. *Educational evaluation and policy analysis* 33, 1 (2011), 95–114.

44. David Williamson Shaffer and Mitchel Resnick. 1999. " Thick" Authenticity: New Media and Authentic Learning. *Journal of interactive learning research* 10, 2 (1999), 195–215.

45. Lee S. Shulman. 1986. Those Who Understand: Knowledge Growth in Teaching. *Educational Researcher* 15, 2 (1986), 4–14.

46. Jeff Sutherland and JJ Sutherland. 2014. *Scrum: the art of doing twice the work in half the time*. Crown Business.

47. Heather Thiry and Sandra L Laursen. 2011. The role of student-advisor interactions in apprenticing undergraduate researchers into a scientific community of practice. *Journal of Science Education and Technology* 20, 6 (2011), 771–784.

48. Heather Thiry, Timothy J Weston, Sandra L Laursen, and Anne-Barrie Hunter. 2012. The benefits of multi-year research experiences: differences in novice and experienced students' reported gains from undergraduate research. *CBE-Life Sciences Education* 11, 3 (2012), 260–272.

49. Bernie Trilling and Charles Fadel. 2009. *21st century skills: Learning for life in our times*. John Wiley & Sons.

50. Etienne Wenger and Jean Lave. 1991. *Situated Learning: Legitimate Peripheral Participation (Learning in Doing: Social, Cognitive and Computational Perspectives) by*. Cambridge University Press, Cambridge, UK.

51. Anna Woodcock, William G Graziano, Sara E Branch, Ida Ngambeki, and Demetra Evangelou. 2012. Engineering students' beliefs about research: Sex differences, personality, and career plans. *Journal of Engineering Education* 101, 3 (2012), 495–511.