Improving engagement, diversity, and retention in computer science with RadGrad: Results of a case study

PHILIP M. JOHNSON and CARLETON MOORE, Department of Information and Computer Sciences, University of Hawaii at Manoa

PETER LEONG and SEUNGOH PAEK, Department of Learning Design and Technology, University of Hawaii at Manoa

Objectives   RadGrad is a curriculum initiative implemented via an application that combines features of social networks, degree planners, individual learning plans, and serious games. RadGrad redefines traditional meanings of "progress" and "success" in the undergraduate computer science degree program in an attempt to improve engagement, retention, and diversity. In this paper, we describe the RadGrad Project and report on an evaluation study designed to assess the impact of RadGrad on student engagement, diversity, and retention. We also present opportunities and challenges that result from the use of the system.

Participants   Our evaluation study involved several major stakeholder groups for undergraduate computer science education at our institution, including: 498 students, 11 Faculty, and 2 Advisors.

Study Methods   We used a qualitative experimental research design, including data obtained through questionnaire responses from our major stakeholder group as well as data from system instrumentation.

Findings   We found strong evidence to support the hypothesis that RadGrad can have a positive impact on student engagement in Computer Science for students who are relatively new to the discipline. We found some evidence to support the hypothesis that RadGrad can have a positive impact on student diversity. We were unable to evaluate the impact of RadGrad on retention.

Conclusions   The data gathered in this study indicates that approaches like RadGrad provide a promising means to improve engagement, diversity, and (potentially) retention in Computer Science. The data suggests that development and evaluation of RadGrad at other institutions and/or in other STEM disciplines would provide useful insight into the generality of the approach. It also indicates that faculty engagement with the technology would increase its effectiveness, but that obtaining faculty buy-in is challenging.

CCS Concepts: • **Social and professional topics** → **Informal education**; **Computing literacy**.

Additional Key Words and Phrases: Curriculum Initiative; Diversity; Engagement; Retention

Authors' addresses: Philip M. Johnson, johnson@hawaii.edu; Carleton Moore, cmoore@hawaii.edu, Department of Information and Computer Sciences, University of Hawaii at Manoa, Honolulu, HI, 96822; Peter Leong, peterleo@hawaii.edu; Seungoh Paek, spaek@hawaii.edu, Department of Learning Design and Technology, University of Hawaii at Manoa, Honolulu, HI, 96822.

1 INTRODUCTION

1.1 Research Problem

Frame the problem or question and its context. Review, critique, and synthesize the applicable literature to identify key issues/debates/theoretical frameworks in the relevant literature to clarify barriers, knowledge gaps, or practical needs.

Traditional curriculum initiatives for undergraduate computer science generally fall into one of three categories. The first involves the injection of a specific technology across the curriculum, such as initiatives for cloud computing [6]. The second involves the injection of a specific domain across the curriculum, such as initiatives for cybersecurity [15] or distributed computing [1]. The third involves the integration of practices intended to address demographic problems including lack of access or diversity, such as initiatives at the University of Illinois [13] and University of Oklahoma [5].

One problem with these types of curriculum initiatives is that they all require changing, well, the curriculum. This is a problem because changing the curriculum is hard: any significant curriculum change will require the approval of committees at multiple levels of the educational institution, it will require buy-in across the faculty, and it will often require significant resources to implement and sustain.

A benefit of changing the curriculum is that if it is accomplished, then *student* buy-in is normally not required. Because the change is to the curriculum, if a student wants to graduate, then they must follow the changed curriculum.

At our University, we have been experimenting with an initiative called "RadGrad", which falls outside these typical categories of curriculum initiatives, and as a result inverts the typical problems and benefits associated with traditional curriculum initiatives.

RadGrad originally rose from two insights: first, a recognition that our department did not have the resources to enable its undergraduate curriculum to keep up with the rate of change in computer science technology and the ever widening domain of application areas, nor could we keep up with student demand for access to curriculum content. In some cases, the local high tech community or online educational services could fill in the gaps, but our department provided no support for student awareness of these extracurricular resources, or guidance as to which ones were worth spending time on, or how such activities might integrate with existing curricular offerings. Finally, since extracurricular activities, are, by definition, "extra", there is a structural incentive for students to limit these activities and instead focus on curricular work which directly affects their GPA.

Second, we recognized that the increase in demand for our undergraduate computer science degree programs was paradoxically leading to a decrease in the diversity of our student body. Increasingly few women and underrepresented groups were making it all the way through our program to graduation. Although approximately 40% of our first year students each year are women, only 15% of our graduating seniors in AY 2018 were women.

In an attempt to address these issues, we created RadGrad, which is a web-based application that combines features of social networks, degree planners, and serious games. Undergraduate students interested in our major can login to RadGrad and declare their interests and career goals. The system then recommends faculty-curated extracurricular opportunities matching their preferences. A degree planner tool enables students to lay out their "degree experience" as a combination of curricular activities (courses) and extracurricular activities (internships, meetups, online courses, hackathons, etc) for each upcoming semester until their planned graduation date. Finally, RadGrad enables students to learn about the broader ways in which computer science impacts upon local, national, and global society, as well as helping them to connect with and create communities of

practice inside and outside of the department. As will be discussed below, related research suggests that making these connections can improve student engagement, diversity, and retention.

To combat the view of extracurricular activities being perceived as "extra", RadGrad does not use GPA to represent student success or progress. Instead, RadGrad provides a three component metric called "myICE", an acronym for Innovation, Competency, and Experience. Students are awarded Competency points for successfully completing a course, and a varying amount of Innovation and/or Experience points for successfully completing an extracurricular activity defined within the system. For example, participating in a hackathon will earn some number of Innovation points, since the goal of hackathons is to create something new. Finishing a summer internship in a high tech company will earn Experience points since that activity provides students with a sense for the demands of a high tech workplace environment. (Depending upon the nature of the internship, it could also earn Innovation points.) To "win" at RadGrad, students must (among other things) earn at least 100 Innovation, Competency, and Experience points by completing some combination of curricular and extracurricular activities by the time they graduate.

Unlike traditional curriculum initiatives, RadGrad exists apart from the official curriculum requirements for our degree programs. As a result, we have been free to deploy and experiment with the approach without the multi-year approval process normally required for curriculum changes. On the down-side, participation in RadGrad is voluntary, and so students must opt in to the system and its philosophy to gain its benefits.

1.2 Research Goals

State the purpose(s)/goal(s)/aim(s) of the study. State the target audience, if specific. Provide the rationale for fit of design used to investigate this purpose/goal (e.g., theory building, explanatory, developing understanding, social action, description, highlighting social practices). Describe the approach to inquiry, if it illuminates the objectives and research rationale (e.g., descriptive, interpretive, feminist, psychoanalytic, postpositivist, critical, postmodern, constructivist, or pragmatic approaches).

RadGrad began in the Fall of 2015 as a student-driven project in several upper-division software development courses. A common perspective from students working on the project at that time was, "I wish I'd had RadGrad when I was starting out in Computer Science". However, most of these students were high achieving, soon-to-graduate individuals who were demonstrating an ability to successfully finish a computer science degree program without the aid of RadGrad. While we found their feedback to be encouraging, we wanted to know whether RadGrad would be interesting and/or useful to students who were more "at risk" in our program. Ideally, we wanted to know if the insights that RadGrad provides regarding the spectrum of computer science careers as well as extracurricular opportunities in the discipline both during and after the degree program would lead to a more diverse and engaged student body, with less attrition over the course of the degree program.

Unfortunately, there have been a multitude of impacts on our computer science degree program over the years of the RadGrad Project, ranging from curricular revisions to changes in advising personnel and process to the biggest impact of them all: COVID-19. Our research approach cannot unambiguously separate the impact of RadGrad on engagement, retention, and diversity from all of these other factors. While we will report on institutional data regarding diversity and retention, our approach focuses on qualitative perspectives from a variety of stakeholders (students, faculty, and advisors) on their view of RadGrad and the use of this data to assess the potential of RadGrad to impact engagement, diversity, and retention, along with insights into how to improve the project in the future.

More specifically, during Academic Year 2021-2022, we conducted a qualitative case study that was designed to provide evidence regarding the following research questions:

- Do students, when introduced to RadGrad, find it of value to their educational experience?
- Do students, after completing their degree program, view RadGrad as having improved their educational experience?
- What are the views of Faculty and Advising staff regarding RadGrad/
- What do these findings imply regarding engagement, diversity, and retention in computer science undergraduate degree programs?

We report on this study starting in Section 4. First, however, we step back to better motivate the design of RadGrad, as well as describe the system in more detail.

2 RELATED WORK

There is a national need for undergraduate computer science degree programs to improve both retention (the percentage of students entering CS programs who finish the degree) and diversity (the percentage of graduates who are female and/or from an underrepresented minority group). We need to improve retention because the projected demand for skills in computer science far exceeds current production [4]. We need to improve diversity because a more diverse STEM population improves tech innovation at large. For example, mixed-sex teams filed 40% more information and technology patents than all-male teams [2], and management diversity leads to a $42M increase in S&P value of firms [7].

While the need is clear, solutions are complicated. Gender diversity in computer science has actually fallen in the last 20 years [8], with no well accepted explanation for its cause. Some diversity-related issues start in middle and high school: black students are less likely than white students to have computer science courses in middle and high school, and female students are less likely than male students to be told they would be good at computer science [9]. There is some research that provides evidence for a way forward: a study by Google [8] concludes that four factors primarily influence young womens' decision to pursue CS: (1) social encouragement (positive reinforcement of CS pursuits from family and peers); (2) self perception (an interest in problem solving and a belief that those skills can be translated to a successful career); (3) academic exposure (availability of curricular and extracurricular CS activities); and (4) career perception (view of CS as a career with diverse applications and a broad potential for positive societal impact). Stout and Camp [14] make similar points around social relevance, a sense of belonging, and cultural bias. RadGrad implements capabilities designed to help address isssues around self perception, academic exposure, career perception, and social relevance among its student users.

For those high school students who graduate and enter an undergraduate degree program in computer science, retention becomes a significant issue. More than half of the students who start out in science or engineering switch to other majors or do not finish college at all [11]. Initiatives to improve retention, such as the Threads undergraduate curriculum at Georgia Tech, emphasize giving students more control over their degree plan, a better understanding of how their studies relate to their career interests, and an increased emphasis on the importance of extracurricular activities [3]. RadGrad provides a technology platform, information system, and incentive structure with these emphases.

Communities of Practice (CoP) is a theory of learning first proposed in 1991 [12], more fully developed in 1998 [16], and extended to "landscapes of practice" in 2004 [17]. A loose definition of Communities of Practice is "groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly." More specifically, three characteristics distinguish a community of practice from other kinds of communities: (1) There is at least one domain of interest shared by all members; (2) members engage in joint activities and discussions,

help each other, and share information; and (3) members are practitioners in the domain, not just people with shared interests, and thus develop a shared repertoire of resources.

Communities of Practice show great promise for improving undergraduate retention and diversity, because participating students will find a new source for social encouragement, self-perception, academic exposure, and career perception. In undergraduate degree programs, CoPs are primarily found within disciplinary-specific extracurricular activities: clubs, meetups, hackathons, and so forth. RadGrad makes these CoPs visible to students, and provides incentives to students for participating in them.

3 RADGRAD

The RadGrad project began in the Fall of 2015. For the first few years, development was student-led as part of various software engineering classes. Development activities during this time focused on the underlying data model, user interface mockups, and possible game mechanics. In 2017, a pilot implementation was deployed into a computer science department and some we began receiving user feedback. In 2018, we began work on RadGrad Version 2, an "industrial strength" version of the system that followed high quality software engineering practices and would enable deployment to different institutions and disciplines. From a technical perspective, RadGrad is now a functional and reliable web-based application, implemented in approximately 30,000 lines of Javascript and 7,000 lines of HTML. It has extensive design and development documentation, unit and integration tests, and implements continuous integration. Instances can be deployed locally or using a cloud-based hosting service such as Digital Ocean. Over 1,000 students have used RadGrad to date.

3.1 Theory of Change

These experiences have helped us understand the "workflow" of RadGrad, and how it implements a theory of change for students, as illustrated by Figure 1.
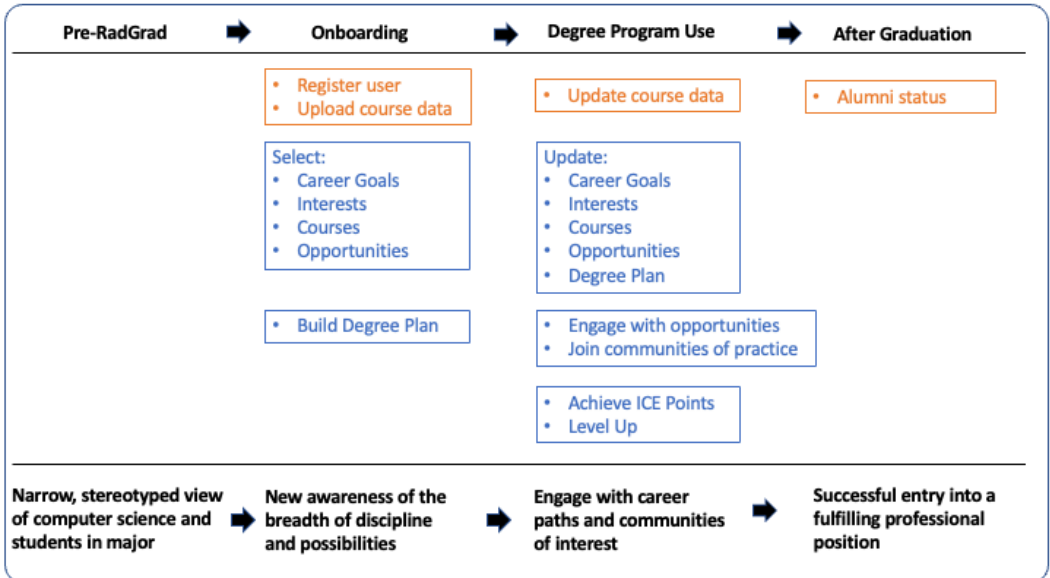

Fig. 1. *RadGrad Theory of Change/Workflow. Orange boxes indicate actions taken by administrators, blue boxes indicate actions taken by students.*

We find that many students in the *Pre-RadGrad* phase have a narrow, stereotyped view of computer science as a discipline (i.e. "it's only about video games") and the students who pursue the major (i.e. "introverted, anti-social white or asian male gamers").

The *Onboarding* phase represents the first exposure of students to RadGrad, and hopefully occurs by the time of their first or second semesters in the degree program. Each RadGrad instance is invitation-only, so an administrator must run scripts to create accounts and upload data on completed degree program coursework. Once registered, students login to the system using their institutional authentication mechanism. We provide an online tutorial [10] that students can follow to familiarize themselves with the system, which teaches them how to select an initial set of Career Goals, Interests, Courses, and Opportunities that they find of interest. RadGrad provides a recommendation system so that once students select Career Goals and Interests, the system can bring related Courses and Opportunities to their attention. The anticipated outcome of the Onboarding process is an increased awareness of the range of disciplinary pursuits, along with concrete activities to pursue to actively explore them.

The *Degree Program Use* phase represents the remainder of the student's degree program, in which they can refer back to RadGrad for new Career Goals and Interests and how they relate to new Courses and Opportunities. RadGrad also includes a Review system that enables students to read about the experiences of other students with Courses and Opportunities. Unlike conventional online systems (like "Rate My Professor"), RadGrad Reviews are not anonymous and are subject to moderation; these two features encourage high quality, thoughtful reviews. The most important goal during Degree Program Use is for students to actually take part in Opportunities, which will connect them with Communities of Practice (both on or off-campus) in a particular disciplinary area. RadGrad provides two game mechanics (myICE Points and Levels) for students who are motivated by those kinds of things.

The *After Graduation* phase represents the state of affairs shortly after matriculation. Our theory of change predicts that students who have taken part in RadGrad will have been more engaged with a wider variety of disciplinary activities occurring both on and off campus, and that those experiences will help them successfully enter into a fulfilling professional position. Once a student graduates, a RadGrad administrator changes their role to "Alumni", which currently converts their account to read-only status.

3.2   From theory of change to user interface

Designing a desirable theory of change is one thing, designing and implementing a usable UI to support it is quite another. We redesigned the RadGrad user interface several times over the past five years based on feedback from students, faculty, and advisors. RadGrad currently provides specialized user interfaces for each of five roles: Student, Faculty, Advisor, Admin, and Alumni. Providing the appropriate UIs for all of these roles requires approximately 130 different web page designs. In this section, we present just a few pages to provide the flavor of RadGrad's user interface.

3.2.1   Home Page/Checklist. We learned from our early prototypes that the "conceptual model" implemented by RadGrad is not intuitively obvious. RadGrad users had problems understanding both what to do and why to do it. We eventually discovered a way to address this "disorientation" by making the Home Page a "smart checklist" with a prioritized list of things to do. Figure 2 illustrates a portion of the home page for a hypothetical student:

Each time this page is retrieved, all of the checklist items associated with the role of the user run code to check various state values and decide whether the checklist item should be displayed as "High Priority", "Medium Priority", or "Completed", as well as what should be included in the content of the checklist item.
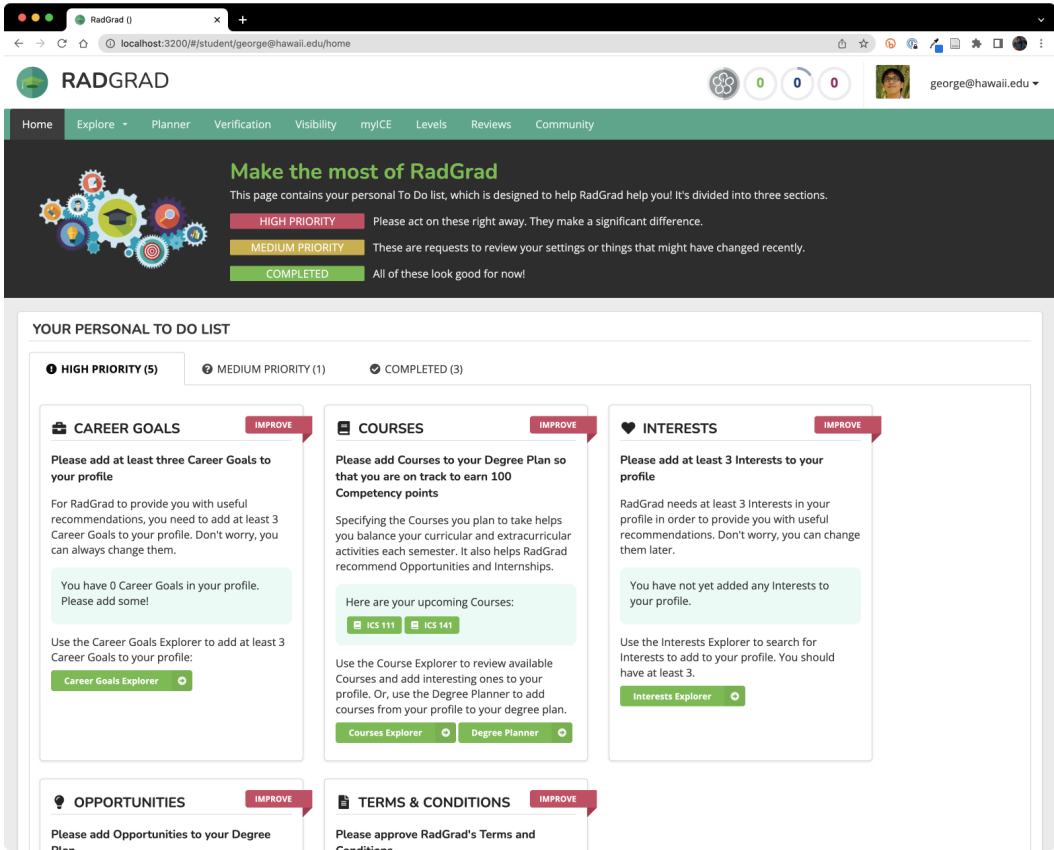
Fig. 2. *RadGrad Home Page (Student Role)*

This checklist design has been well received by users, because it provides direct, personalized feedback on both what to do next (if anything), as well as why to do it. Checklists are not only useful when a user is first learning to use RadGrad, but is also helpful as a means to inform users of new content in the system or other changes.

Finally, checklists are helpful not just to students! We find that faculty, advisors, and even administrators benefit from a smart, personalized checklist home page.

3.2.2 Content Explorer Pages. The current version of RadGrad for computer science at our institution defines 20 Career Goals, 78 Interests, 57 Opportunities, and 106 Courses. This is a significant amount of inter-dependent content. RadGrad provides "Explorers" for each of these four content types that enable users to scroll through brief summaries of each item.

It is straightforward to search the Internet for "Computer Science Careers" (for example) and find a plethora of pages. RadGrad provides unique value to students by providing relationships between content items and users, as illustrated in Figure 3.

This page shows that for this (hypothetical) student user, (a) they have added Data Scientist to their profile as a Career Goal; (b) Of the five Interests associated with the Data Scientist Career Goal, the student has two of them in their profile (Algorithms and Databases); (c) The student has completed six Courses related to Data Science already; (d) The student has already completed three

Fig. 3. *RadGrad Data Scientist Details Page*

Opportunities related to Data Scientist, has one upcoming, and there are quite a few more related Opportunities that could be added to their plan; (e) Six students in this community have Data Scientist in their profile (and have opted in to make this information visible to other registered users), and (f) six faculty have listed Data Scientist as a Career Goal for which they could provide advice and guidance. All of this information is in addition to a brief profile of the Career Goal along with a "Day in the Life" video.

It is important to note that RadGrad content is "faculty-curated". It is the responsibility of one or more faculty in the discipline at the institution to determine what constitutes relevant Career Goals, Interests, and Opportunities for their students, and how these items are documented in the system. There is no central authority decreeing content. This enables a RadGrad instance to be appropriately tailored to the educational context, but implies faculty participation.

3.2.3   The Degree Plan Page. As part of the onboarding process, students peruse RadGrad content and add relevant Career Goals, Interests, Opportunities, and Courses to their "profile". Once a student has identified *what* they are interested in, the next step is to determine *when* they want to participate. Answering the latter question is the function of the Degree Planner, as illustrated in Figure 4.

The left side of the Degree Planner contains a grid representing a set of academic years, each containing three semesters (Fall, Spring, Summer[1]. For each of the semesters, student indicates not just their CS-related curricular activities (i.e. Courses) but also their CS-related extracurricular activites (called "Opportunities" in RadGrad).
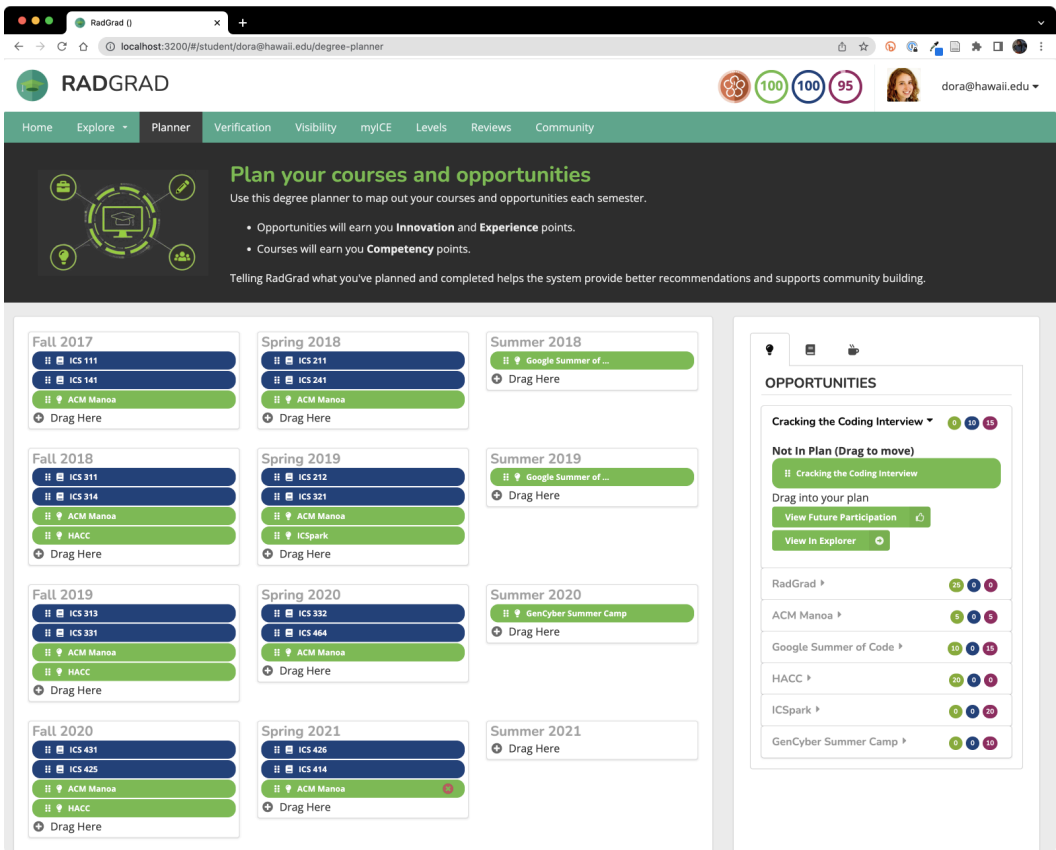
Fig. 4. *RadGrad Degree Plan Page (Student Role)*

The Degree Planner has a pane on the left side that shows the semester in which a student plans to take advantage of a Course and/or Opportunity. The right side provides access to the Courses and Opportunities in the student's profile. By expanding an item (such as the "Cracking the Coding Interview" Opportunity), a tile is revealed that can be dragged to any current or future semester.

3.2.4   Levels and myICE: What's good enough? Figure 5 zooms in on the portion of the navigation bar containing a student's Level (a green RadGrad icon) and myICE Points (three circles containing

---

[1]RadGrad can be configured to support institutions on the quarter system

Fig. 5. *UI for student Level and myICE Points*

numbers inside). RadGrad provides a game mechanic consisting of six "Levels", each represented by a different colored RadGrad icon. As students earn myICE Points and carry out other RadGrad activities, such as writing reviews of courses and opportunities, they can progress through white, yellow, green, blue, brown, and black levels, similar to martial art belt colors. The rules for achieving Levels are designed such that a motivated student can achieve a new Level each semester, so a total of six semesters (typically three years) is required to get through all Levels. We print laptop stickers for each Level, and students can get a new laptop sticker free from the department with the color corresponding to each Level as they achieve it. This provides visibility for RadGrad and for the student's participation. The student in Figure 5 is currently at Level 3 (green).

We developed a circular representation for myICE Points to help students quickly understand their progress using RadGrad. Achieving a successful undergraduate degree experience according to RadGrad's game mechanics means that a student must first plan out a sequence of curricular and extracurricular activities that will eventually earn them at least 100 Innovation, Competency, and Experience points. Once they have made that plan (which they are free to change at any time), they must then actually earn the points by completing the curricular and extracurricular activities. To verify completion of a curricular activity, RadGrad imports data from the University's course management system indicating the coursework the student has taken and the grade they received. To verify completion of an extracurricular activity, the student must first "request verification" of the Opportunity. This request sends a message to the RadGrad administrator or faculty member who has responsibility for checking that the student participated. The admin or faculty member can accept or decline the verification request, and if accepted, the system will award the points associated with the Opportunity to the student.

RadGrad represents myICE by three circles: a green circle for Innovation, a blue circle for Competency, and a red circle for Experience. For each of the three components of myICE, the circle is partially or fully colored in a light shade to represent the number of *planned* points, based upon what the student has put in their degree plan. In Figure 5, the Innovation circle is colored only half way in a light green color, which indicates that this student has only planned to earn 50 Innovation points so far. The blue circle is completely filled in, indicating that the student has planned to achieve 100 Competency points. Finally, the Experience circle is also only half filled, indicating that only 50 Experience points are currently in the degree plan.

RadGrad uses a dark shade on the circle to represent the number of points actually earned so far, and also indicates that number in the center of the circle. So, the student in Figure 5 has so far earned 5 Innovation, 78 Competency, and 5 Experience points.

Finally, RadGrad defines four other roles for users in addition to Student: Admin, Faculty, Advisor, and Mentor. The user interface to RadGrad varies significantly in functionality depending upon the role of the user. For example, users with the Advisor role have the ability to add new students to the system, and see existing student degree plans. Due to space limitations, only a brief overview of functions and capabilities associated with the Student role are described in this section. For more detailed information about the system, please go to https://radgrad.org.

3.3   From user interface to user experience

Finally, RadGrad defines four other roles for users in addition to Student: Admin, Faculty, Advisor, and Mentor. The user interface to RadGrad varies significantly in functionality depending upon the role of the user. For example, users with the Advisor role have the ability to add new students to the system, and see existing student degree plans. Due to space limitations, only a brief overview of functions and capabilities associated with the Student role are described in this section. For more detailed information about the system, please go to https://radgrad.org.

Finally, RadGrad defines four other roles for users in addition to Student: Admin, Faculty, Advisor, and Mentor. The user interface to RadGrad varies significantly in functionality depending upon the role of the user. For example, users with the Advisor role have the ability to add new students to the system, and see existing student degree plans. Due to space limitations, only a brief overview of functions and capabilities associated with the Student role are described in this section. For more detailed information about the system, please go to https://radgrad.org.

Finally, RadGrad defines four other roles for users in addition to Student: Admin, Faculty, Advisor, and Mentor. The user interface to RadGrad varies significantly in functionality depending upon the role of the user. For example, users with the Advisor role have the ability to add new students to the system, and see existing student degree plans. Due to space limitations, only a brief overview of functions and capabilities associated with the Student role are described in this section. For more detailed information about the system, please go to https://radgrad.org.

4   METHOD

4.1   Research Design Overview

Summarize the research design, including data-collection strategies, data-analytic strategies, and, if illuminating, approaches to inquiry (e.g., descriptive, interpretive, feminist, psychoanalytic, postpositivist, critical, postmodern, constructivist, or pragmatic approaches). Provide the rationale for the design selected.

4.2   Study Participants

4.2.1   Researcher Description. Describe the researchers' backgrounds in approaching the study, emphasizing their prior understandings of the phenomena under study (e.g., interviewers, analysts, or research team). Describe how prior understandings of the phenomena under study were managed and/or influenced the research (e.g., enhancing, limiting, or structuring data collection and analysis).

4.2.2   Participants. Provide the numbers of participants/documents/events analyzed. Describe the demographics/cultural information, perspectives of participants, or characteristics of data sources that might influence the data collected. Describe existing data sources, if relevant (e.g., newspapers, internet, archive). Provide data repository information for openly shared data, if applicable. Describe archival searches or process of locating data for analyses, if applicable.

4.2.3   Researcher–Participant Relationship. Describe the relationships and interactions between researchers and participants relevant to the research process and any impact on the research process (e.g., was there a relationship prior to research, are there any ethical considerations relevant to prior relationships).

4.3   Participant Recruitment

4.3.1   Recruitment Process. Describe the recruitment process (e.g., face-to-face, telephone, mail, email) and any recruitment protocols. Describe any incentives or compensation, and provide assurance of relevant ethical processes of data collection and consent process as relevant (may include institutional review board approval, particular adaptations for vulnerable populations, safety monitoring). Describe the process by which the number of participants was determined in relation to the study design. Provide any changes in numbers through attrition

and final number of participants/sources (if relevant, refusal rates or reasons for dropout). Describe the rationale for decision to halt data collection (e.g., saturation). Convey the study purpose as portrayed to participants, if different from the purpose stated.

4.3.2   Participant Selection. Describe the participants/data source selection process (e.g., purposive sampling methods, such as maximum variation; convenience sampling methods, such as snowball selection; theoretical sampling; diversity sampling) and inclusion/exclusion criteria. Provide the general context for the study (when data were collected, sites of data collection). If your participant selection is from an archived data set, describe the recruitment and selection process from that data set as well as any decisions in selecting sets of participants from that data set.

4.4   Data Collection

4.4.1   Data Collection/Identification Procedures. State the form of data collected (e.g., interviews, questionnaires, media, observation). Describe the origins or evolution of the data-collection protocol. Describe any alterations of data-collection strategy in response to the evolving findings or the study rationale. Describe the data-selection or data-collection process (e.g., were others present when data were collected, number of times data were collected, duration of collection, context). Convey the extensiveness of engagement (e.g., depth of engagement, time intensiveness of data collection). For interview and written studies, indicate the mean and range of the time duration in the data-collection process (e.g., interviews were held for 75 to 110 min, with an average interview time of 90 min). Describe the management or use of reflexivity in the data-collection process, as it illuminates the study. Describe questions asked in data collection: content of central questions, form of questions (e.g., open vs. closed).

4.4.2   Recording and Data Transformation. Identify data audio/visual recording methods, field notes, or transcription processes used.

4.5   Analysis

4.5.1   Data-Analytic Strategies. Describe the methods and procedures used and for what purpose/goal. Explicate in detail the process of analysis, including some discussion of the procedures (e.g., coding, thematic analysis) following a principle of transparency. Describe coders or analysts and their training, if not already described in the researcher description section (e.g., coder selection, collaboration groups). Identify whether coding categories emerged from the analyses or were developed a priori. Identify units of analysis (e.g., entire transcript, unit, text) and how units were formed, if applicable. Describe the process of arriving at an analytic scheme, if applicable (e.g., if one was developed before or during the analysis or was emergent throughout). Provide illustrations and descriptions of the analytic scheme development, if relevant. Indicate software, if used.

4.5.2   Methodological Integrity. Demonstrate that the claims made from the analysis are warranted and have produced findings with methodological integrity. The procedures that support methodological integrity (i.e., fidelity and utility) typically are described across the relevant sections of a paper, but they could be addressed in a separate section when elaboration or emphasis would be helpful. Issues of methodological integrity include the following: Assess the adequacy of the data in terms of its ability to capture forms of diversity most relevant to the question, research goals, and inquiry approach. Describe how the researchers' perspectives were managed in both the data collection and analysis (e.g., to limit their effect on the data collection, to structure the analysis). Demonstrate that findings are grounded in the evidence (e.g., using quotes, excerpts, or descriptions of researchers' engagement in data collection). Demonstrate that the contributions are insightful and meaningful (e.g., in relation to the current literature and the study goal). Provide relevant contextual information for findings (e.g., setting of study, information about participants, interview question asked is presented before excerpt as needed). Present findings in a coherent manner that makes sense of contradictions or disconfirming evidence in the data (e.g., reconcile discrepancies, describe why a conflict might exist in the findings). Demonstrate consistency with regard to the analytic processes (e.g., analysts may use demonstrations of analyses to support consistency, describe their

development of a stable perspective, interrater reliability, consensus) or describe responses to inconsistencies, as relevant (e.g., coders switching midway through analysis, an interruption in the analytic process). If alterations in methodological integrity were made for ethical reasons, explicate those reasons and the adjustments made. Describe how support for claims was supplemented by any checks added to the qualitative analysis. Examples of supplemental checks that can strengthen the research may include: transcripts/data collected returned to participants for feedback triangulation across multiple sources of information, findings, or investigators; checks on the interview thoroughness or interviewer demands; consensus or auditing process; member checks or participant feedback on findings; data displays/matrices; in-depth thick description, case examples, or illustrations; structured methods of researcher reflexivity (e.g., sending memos, field notes, diary, logbooks, journals, bracketing); checks on the utility of findings in responding to the study problem (e.g., an evaluation of whether a solution worked)

5 FINDINGS

Describe research findings (e.g., themes, categories, narratives) and the meaning and understandings that the researcher has derived from the data analysis. Present research findings in a way that is compatible with the study design. Demonstrate the analytic process of reaching findings (e.g., quotes, excerpts of data). Present synthesizing illustrations (e.g., diagrams, tables, models), if useful in organizing and conveying findings. Photographs or links to videos can be used.

6 DISCUSSION

6.1 Central Contributions to the Discipline

Describe the central contributions and their significance in advancing disciplinary understandings.

6.2 Types of Contributions Made by Findings

Describe the types of contributions made by findings (e.g., challenging, elaborating on, and supporting prior research or theory in the literature describing the relevance) and how findings can be best utilized.

6.3 Comparing Prior Theories and Research Findings

Identify similarities and differences from prior theories and research findings.

6.4 Alternative Explanations of Findings

Reflect on any alternative explanations of the findings.

6.5 Strengths and Limitations

Identify the study's strengths and limitations (e.g., consider how the quality, source, or types of the data or the analytic processes might support or weaken its methodological integrity).

6.6 Limitations on Transferability

Describe the limits of the scope of transferability (e.g., what should readers bear in mind when using findings across contexts).

6.7 Ethical Dilemmas/Challenges

Revisit any ethical dilemmas or challenges that were encountered, and provide related suggestions for future researchers.

6.8 Implications for Future

Consider the implications for future research, policy, or practice.
CS Education for the Socially-Just worlds we need

Confronting inequities: a case for critical theory
intersectional experiences of black women in computing
What is the theory of change for faculty? for administrators?

7 ACKNOWLEDGEMENTS

REFERENCES

[1] Michael Abebe, Brad Glasbergen, and Khuzaima Daudjee. 2019. WatDFS: A Project for Understanding Distributed Systems in the Undergraduate Curriculum. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 920–926. https://doi.org/10.1145/3287324.3287473 event-place: Minneapolis, MN, USA.

[2] Catherine Ashcraft and Anthony Breitzman. 2012. *Who invents it? Women's participation in information technology patenting.* Technical Report. National Center for Women and Information Technology. http://www.ncwit.org/sites/default/files/resources/2012whoinventsit_web_1.pdf

[3] Jerri Barrett. 2017. Expanding the Pipeline: Key Learnings on Retaining Underrepresented Minorities and Students with Disabilities in Computer Science. https://goo.gl/CQfy2K

[4] Tracy Camp. 2017. *Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006.* Technical Report. Computing Research Association. https://cra.org/wp-content/uploads/2017/02/Generation-CS.pdf

[5] Mathilde Collain and Deborah Trytten. 2019. "You Don'T Have to Be a White Male That Was Learning How to Program Since He Was Five". In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 968–974. https://doi.org/10.1145/3287324.3287383 event-place: Minneapolis, MN, USA.

[6] Debzani Deb, Muztaba Fuad, and Keith Irwin. 2019. A Module-based Approach to Teaching Big Data and Cloud Computing Topics at CS Undergraduate Level. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 2–8. https://doi.org/10.1145/3287324.3287494 event-place: Minneapolis, MN, USA.

[7] Cristian Dezso and David Ross. 2007. *Girl Power: Female participation in top management and firm performance.* Technical Report. University of Maryland. http://bit.ly/2WZH1fa

[8] Hai Hong and Abby Bouchon. 2014. *Women who choose computer science: what really matters.* Technical Report. Google, Inc. http://bit.ly/2x8r8nC https://static.googleusercontent.com/media/edu.google.com/en//pdfs/women-who-choose-what-really.pdf.

[9] Google Inc. 2016. *Diversity Gaps in Computer Science: Exploring the Under-representation of Girls, Blacks and Hispanics.* Technical Report. Google, Inc. https://services.google.com/fh/files/misc/diversity-gaps-in-computer-science-report.pdf

[10] Philip Johnson. 2022. *RadGrad New Student Tutorial.* Technical Report. University of Hawaii. https://www.radgrad.org/docs/users/new-student/overview

[11] Nancy Kober. 2015. *Reaching Students: What Research Says About Effective Instruction in Undergraduate Science and Engineering.* National Academies Press. https://doi.org/10.17226/18687

[12] Jean Lave and Etienne Wenger. 1991. *Situated Learning: Legitimate peripheral participation.* Cambridge University Press.

[13] Heather E. Metcalf, Tanya L. Crenshaw, Erin Wolf Chambers, and Cinda Heeren. 2018. Diversity Across a Decade: A Case Study on Undergraduate Computing Culture at the University of Illinois. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 610–615. https://doi.org/10.1145/3159450.3159497 event-place: Baltimore, Maryland, USA.

[14] Jane Stout and Tracy Camp. 2014. Now what? Action items from social science research to bridge the gender gap in computing research. *SIGCAS Computers and Society* 44, 4 (Nov. 2014). https://cra.org/cerp/wp-content/uploads/sites/4/2015/09/Stout-Camp-2015.pdf

[15] Cara Tang, Cindy S. Tucker, Christian Servin, Markus Geissler, and Melissa Stange. 2019. Shaping Curricular Guidelines for Associate-Degree Cybersecurity Programs. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 334–335. https://doi.org/10.1145/3287324.3287516 event-place: Minneapolis, MN, USA.

[16] Etienne Wenger. 1998. *Communities of Practice: Learning, Meaning, and Identity.* Cambridge University Press. Google-Books-ID: Jb8mAAAAQBAJ.

[17] Etienne Wenger, Mark O'Creevy, Steven Hutchinson, Chris Kubiak, and Beverly Wenger-Traynor. 2004. *Learning in landscapes of practice.* Routledge.