CPATH CB: Building a Platform for Learning – A Learner Centered Approach to Computer Science Education

1. Introduction

Despite tremendous advances in the science of computing, and the revolutionary effect it has had on our society and just about every aspect of daily life over the past two decades, computer science education has largely remained unchanged [6]. While the content of lectures and courses evolves to incorporate and reflect current practices and theories, the way we deliver said content and structure students' education has largely remained the same. Despite being one of the most rapidly evolving disciplines, we have struggled to incorporate innovation and new ideas into our educational programs, especially when these do not fit into existing courses. As a consequence, a growing proportion of our population is inadequate prepared to fully contribute to our technologically advanced society [6]. If we are to maintain our leadership position in the technological and economic spheres, it is essential that we change the way we teach computing in order to reach a larger, and more diverse population.

The perception of stagnation and lack of diversity is a negatively reinforcing trend; non-traditional students are driven away. In a time when we should be seeing a broadening of our demographics brought on by a growth in opportunities and interdisciplinary use of computing in other fields and disciplines, we instead find a narrowing of interest and decreased diversity in our student population [6, 11]. This is a troubling development. A smaller and less diverse student population means that we are less equipped to meet the needs of increasingly computer-dependent industries in the U.S., struggling to retain their global competitiveness [2].

In order to break this cycle we need to radically change the nature of computer science education, not just the content of courses, but the structure of both our programs and the student experience. Among our challenges is the need to find ways of presenting relevant material in ways more appropriate for a more diverse population, find better ways to reach and recruit new student populations, and find better ways to identify, promote, and reward skills and traits such as initiative, leadership, innovation, creativity, collaboration, and interdisciplinary work. If successful, we will also need to deal with the challenge of teaching an increasingly diverse population, not only culturally, but also in terms of their background, skills, and abilities. We need to ensure that we can still challenge the talented and motivated traditional students, while supporting and nurturing less traditional population groups. In order to examine and identify potential answers to all these challenges we need to bring together a diverse research and educational team.

The purpose of this proposal is therefore to enable us to (1) bring together and build a community of partners and stakeholders which will help us identify and institute meaningful change, and (2) develop, prototype, evaluate, and refine a set educational, organizational, and cultural changes to our program. This grant will enable us to not only build a community of educators, researchers, and industry representatives from different areas and disciplines, but also help us develop a model and detailed implementation and evaluation plan to be shared with other institutions as part of a future transformational grant and other collaboration and dissemination activities.

1.1 Vision

Our vision of how to redesign Computer Science (CS) education is based in part on our experiences redesigning our Electrical and Computing Engineering (ECE) program, applying a learner centered integrative approach to teaching. We also leverage the feedback we have already collected from different constituents, and reform ideas under consideration or being implemented.

Our view is that meaningful curriculum change must be driven by the needs of our constituents; employers, students, as well as educators and evaluators. Reform must reflect the needs of our constituents, or desired constituents, and it must be the product of a sustainable iterative process of benchmarking, problem identification, vetting ideas and feedback, and prototyping solutions, following the Design Research model [12]. This is essential in order to avoid returning to the situation we find ourselves in today, and will require a radically different organizational model than what we have today.

Rather than depending on yearly advice from our industry advisory board, we envision bringing together an active community of advisors from industry, community colleges, the local community, former graduates, current students, and faculty on a quarterly basis to provide targeted, specific guidance to the project team. This community of advisors, trusted with both success and failure stories, will become an essential component to the program, and our redesign process, and will help us communicate with the larger constituent community.

We envision a curriculum based around unifying threads, giving student curriculum continuity and reinforcing learning. Potential threads, derived from initial feedback from our industry advisory board and community leaders include multi core programming, graphics and simulation, software engineering principles, Open Source development and professional skills including leadership, oral and written communications, and innovation skills. These threads will serve to tie classes together by leveraging common applications, projects, examples, and case-studies, an approach which has proved highly successful in our ECE program.

We also envision a program based around a *Community of Code* [1, 7]. The Community of Code is part Open Source repository [13, 16] and part online community [4]. The Community of Code serves four purposes; (1) it serves as repository and facilitator for student-led, hands-on learning and projects, (2) it serves as social networking and peer support tool, (3) it brings the students into contact with the rest of the academic community and industry by hosting and promoting interdisciplinary community projects, and (4) it serves as a showcase for both recruiters and potential students, offering a more realistic view of the interdisciplinary and creative nature of computer science.

By building a controlled setting for students to collaborate and showcase projects and ideas, we help motivate them, support creativity, initiative, innovation, and collaboration, in accordance with project based [3] and constructivist learning theory [4, 8]. This of course needs to be done in a controlled fashion in order to avoid some of the problems associated with the Open Source model, including high levels of competition [14]. By building and stressing the community aspects, we aim to help students, especially non-traditional students build stronger social networks with their peers, breaking the sense of isolation and providing motivation for participation. By bringing in the larger community in as project partners and sponsors, we encourage students to form interdisciplinary teams, encourage non-CS students to get involved in our programs, and allow sponsors to propose projects which will appeal to a wider student base.

1.2 Goals & Objectives

Our primary goal for this project is to bring together a strong and active community of advisors to guide change, while overcoming coordination, communication, trust, and administrative challenges posed by the close collaboration we envision. In addition, we have already started to pilots changes to our program based on early feedback and observations. Our goal is to evolve and evaluate these changes as part of the grant activities. These activities include:

- **Build a community of constituents** to advise and guide our efforts to reinvigorate and redesign our CS program and curriculum, including more up-to-date materials and

curriculum continuity. Validate this model of collaboration and governance as a working model for continuous program governance, evaluation, and improvement. Include the following constituents in the community of advisors:
    - Industry partners; recruit representatives from leading local, national, and international employers and visionaries, including HP, IBM, Intel, and Nvidia to name a few.
    - Community and local colleges; a major source (40%) of incoming students for our program.
    - Local high school teachers and career advisors
    - Community and non-profit representatives, including the National Center for Women and IT (NCWIT)
    - Alumni groups
    - Undergraduate students
    - Faculty, from inside the school, other departments, and other universities

- **Build a Community of Code** designed to bring students together, breaking the isolation and disconnectedness experienced by many students, especially underrepresented groups. Additionally, it offers more opportunities for hands-on activities and personal ownership in order to build a culture of innovation, initiative, collaboration, and leadership and it Offers more opportunities for interdisciplinary and community driven work, giving students opportunities to see computing in a broader context, and the applicability of their skills. For the student community centered on the Community of Code, our gaols and objectives are:
    - Build, deploy, and evaluate the adoption and use of such communities with regards to the creation of peer-support networks, and retention rates among incoming students
    - Determine the impact of bringing external groups into the community as sources for project ideas and mentorship.
    - Evaluate the projects potential as a tool for changing public perception about computer science as a discipline among prospective students.
- **Leverage existing outreach efforts** by including representatives from these groups as part of the community of advisors. Work with them to customize their efforts to improve the recruiting into CS programs and improve the overall image of the discipline as a whole.
- **Work closely with the larger community of constituents**, as well as state and national partners to develop the foundation for a replicable model, to be explored in a future transformational grant.

1.4 Outcomes

After completing this project, we will have:

- Established an ongoing relationship with a panel of advisers, working in partnership with the School of EECS to provide detailed guidance, with preliminary data indicating whether this is an effective model of governance.
- Developed a clear and verified plan for reworking the way we teach CS, focusing primarily on changing the way we teach as well as addressing the content of classes.
- Created a detailed evaluation plan and benchmark data which will allow us to measure the impact and success of the proposed program.
- Received extensive feedback and evaluated ideas based on our experiences that can be communicated to a national audience of schools. (This effort is part of another CPATH CB proposal that other faculty at OSU are participating in with Robert Schnabel at the

University of Colorado.)

2. Implementation Plan

The need to rethink and reinvigorate CS education is a need which we have already recognized, and a goal towards which we have been working for some years. In this section we give a brief overview of some of the more relevant activities we have engaged in, along with what observations we have as to their effectiveness. We then discuss in detail our proposed community building plans and activities, how we envision this community driving change, and our plans for next steps.

2.1 Leveraging Previous Curriculum Development Activities

Key among our previous activities in rethinking redesigning educational models and practices is leveraging our work redefining our ECE curriculum and student experiences. (Note: The ECE and CS departments merged nearly 4 years ago to help facilitate collaborative research and education efforts.) This effort involved a dedicated community including industrial representatives intimately involved in changes, current students providing timely input and help with curricular modifications, faculty and staff (both in the discipline and from other departments) passionate about student learning and experiences. Over a 5 year period, we have transformed the program by convening weekly meetings, seeking outside industry input through regular industry visits, involving community college faculty providing supportive development and feedback, as well as interaction with departments outside the typical sphere of involvement such as art. The result of this work was something we call a *Platform for Learning* [17], a coherent set of experiences and social structures that unify and reinforce the concepts introduced in the undergraduate curriculum.

Beginning in the freshman year, each ECE student constructs a basic robot, we call *TekBots*®, that they build on as they progress through their four-year program. By doing this, the subject areas become connected and students have a context for their learning similar to what practicing engineers experience in their profession. As students take courses throughout the program, they apply their knowledge to this platform so that it becomes a representation of what they have learned and how the various topics they have learned are connected. To illustrate the benefits of this platform, we evaluated the introduction of the platform in the freshmen digital logic class. Using the platform, students' ability to generate new ideas and also work as part of a community of learners was assessed and it was determined that both of these areas showed increases [10]. More than 9 classes have been transformed in the freshmen through senior year of the curriculum with the first group of students with the full four year curriculum graduating last year. Faculty working with these students in graduate school are finding they have superior understanding of the fundamental theory as well as effective hands-on application of the ECE fundamentals.

To date, the platform for learning has been used by more than 2000 students at Oregon State University, Texas A&M University, University of Nebraska (Omaha), Fukuoka Institute of Technology (Japan), and Linn-Benton Community College. Additionally, the program has received commitments in time and money from companies including Tektronix, Analog Devices, National Semiconductor, Texas Instruments, Atmel, and Agilent Technologies.

The key foundations of the Platform for Learning approach include:

- Community building – Students are encouraged to and given the tools and contexts to interact with the larger community, both within and outside the department, building social and support networks, as well as collaborations which will last them throughout and beyond their college years.
- Curriculum continuity and integration – The platform provides clear continuity and

progression of ideas and concepts throughout the course-work. Additionally, it clearly demonstrates how the various topics in the curriculum are inter-related.

- Hands-on minds-on contextual learning – Students are shown practical applications of theoretical ideas by working on real-world applications and real-world problems, problems which are personally meaningful and engaging to the student.
- Personal ownership – Students create their own "platform", working on projects which are personally meaningful, and which they can customize to their interests and goals.

A platform for learning complements the existing structure of lecture and lab. It makes lecture topics more concrete and active and meshes them with lab experiences. It acts to expand and integrate the entire curriculum as illustrated in Fig. 1. Class lectures are effective in providing depth and breadth in the discipline. This knowledge introduction is complemented by the laboratory where hands-on experiences reinforce the lecture material. Integrating a platform for learning into the curriculum expands the learning opportunities and effectiveness in many different dimensions as illustrated in Fig. 1. With a platform integrated into the class, a stronger sense of community emerges with personal ownership for their learning yet clearly motivated by the opportunities to explore knowledge in a fun environment. Because the educational experiences build on one another, students gain a clear understanding of how topics inter-relate, how to find solutions to things that "go wrong" and they develop their innovation skills in the process.

Our intent through this grant is to leverage this knowledge and experience in building a strong community committed to curriculum innovation and thus building from our previous experiences with curriculum development, implementation, and evaluation. Because of the differences between Electrical and Computer Engineering and Computer Science, a different platform for learning is appropriate, but many of the fundamental and pedagogical principles directly apply. The platform for learning we intend to consider as part of this effort is based on a community of code, a combination code repository and online community.

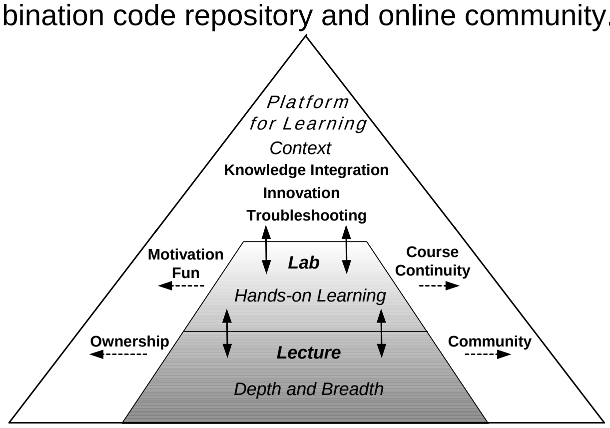

Figure 1: A platform for learning expands the learning opportunities by providing context, knowledge integration, innovation and problem solving experiences. It also enhances ownership, motivation, community, and course continuity.

2.2 Building an Active Community of Advisors

The key component of this proposal is about bringing together a community of constituents, partners, and stakeholders of our program to evaluate, discuss, and propose real and meaningful change to how and what we teach in computing. Rather than engaging a nationwide audience, we seek to establish a close, highly active partnership aimed at helping us derive and refine a new model for CS education. Our goal is to build a model based on close collaboration with all constituent groups, and determine if it can be a successful model before we engage a nationwide audience.

Our goal therefore is to build a constituent community, composed of approximately 15-20 stakeholders, representing all the groups which depend on us, or on whom we depend. This group, unlike the advisory panels commonly seen, and already existent here, is meant to assume a more direct and hands-on responsibility for the program. Our goal is to put together a group which will work directly with faculty and students, meeting on a quarterly basis, to reshape our program. This community of advisors will support the project team by reviewing the work done between quarterly meetings. The project team will consist of the PIs and Co-PIs, graduate students and undergraduate students working on the project on a daily basis.

This feedback, and the ideas derived from such an active constituent community is essential to determining the direction and requirements for a revised educational program. These groups include the students themselves (past and current), major employers and trendsetters, our peers, and our sources for students (and therefore the ones preparing our students before they come here). Without active input and support from these groups, changes which we make here will likely fall flat, or be less effective.

The suggestions and ideas derived from this committee of advisors, properly vetted to ensure they meet the requirements of the university and our accreditation, will be presented to various other groups that are currently active. We will identify key representatives from these groups to be on the community of advisors. Additionally, we will seek overall feedback from these groups on a yearly basis. The constituent groups include the following:

- Other departments & programs (within Oregon State University, or the University System of Oregon), as partners in the community building efforts, to help us identify potential pedagogical approaches and ideas which our CS educated constituents may be unfamiliar with, and to help in the evaluation of our efforts. Specifically we will seek to recruit educators and researchers from the department of Science and Mathematics Education who have extensive experience with these activities, as well as representatives from psychology and the liberal arts addressing the critical communication skills needed by our students. (2-3 people)
- Community and non-profit representatives, including the National Center for Women and IT (NCWIT). (1-2 people)
- Community colleges – This is a well established group that meets quarterly to discuss and coordinate efforts in the state. Currently, 40% of our CS students transfer from community colleges in Oregon. It is critical that our programs and reforms are adopted or at least compatible is required to meet agreements established at the state level for clear matriculation of transfer students. (1-2 people)
- High-schools, our traditional source of students. Work with existing outreach efforts (see later sections) to try and reach more non-traditional students through teachers and advisors. (1-2 people)
- Industry and professional group representatives of major employers and leaders nationally or regionally (5-7 people). Commitments have already been made by several of our strong supporters including Rajeev Panday (HP-Software Engineer and Alum), Shane Wall (Intel – VP and Alum), Dan Frye (IBM-Open Source Lab) and J.J. Cadiz (Microsoft Labs and Alum).
- Alumni – Many of our professional representatives and community college faculty are alums of Oregon State so no others will be added to the community of advisors. However, it is important to recognize that there will be significant representation from this group.
- Current undergraduate students – Through the ECE curriculum develop, we have found that our current undergraduates are among the best at providing real, relevant and insightful input to our program. We will select a representative group of leaders to further our efforts. (2-4 students at varying levels of the program)

- Oregon Department of Education - Tom Thompsen who has been involved with the ECE curriculum development as part of his Ph.D. in Education recently joined the Oregon Department of Education. He is excited to help further this effort and connect it with K-12 education. (1 individual)

These groups represent the major stakeholder groups for our program identified at this time. The numbers presented in the preceding list are a rough guide only, we will make necessary adjustments as needed if we determine that groups are significantly over or underrepresented. As mentioned earlier, and as will be elaborated in our project plan, our goal is to establish a working group which will be able to work effectively together and ensure that innovative ideas surface. We also need to make sure the group is big enough and flexible enough to deal with the amount of work envisioned, and to survive temporary fluctuations in schedules and membership over the lifetime of the project.

2.3 Driving Curriculum Change

The purpose of forming the community of advisors is to drive rapid, meaningful curriculum change. Though the community does not yet exist, we have a long history of experimenting and reforming curriculum. Therefore, we have already collected feedback and ideas from some of the groups which will form part of our constituent community, and as described in Section 2.1, have already engaged in some reform efforts, in part through a grant from Google, aimed at helping us re-examine our curriculum and better incorporate Open Source development and Open Source ideas.

The constituent community therefore will not be starting from scratch when formed, it will have a detailed set of data-points, ideas, and preliminary results presented to it, and will be able to determine if those efforts go forward, are modified or scaled back, or are abandoned outright. The following section presents our current vision for a reformed curriculum and curriculum design process, though the details are in some cases intentionally left vague awaiting specific guidance from our future constituent community, or evaluation results.

In the next sections we outline our plan for course evolution, for ensuring curriculum continuity and program cohesion, and our second community building effort, aimed at building a community of and for our undergraduate students.

2.3.1 Curriculum Transformation

Based on the feedback derived from our advisory board we must be prepared to engage in an iterative design research process [12] to redesign the curriculum and degree structure. In addition to taking into account the feedback from our stakeholder community, we must also determine students' reactions and feedback, as well as the requirements put in place by the university and accrediting bodies like ABET. Students are our best indicators of how the process is progressing, and their opinion must be taken seriously.

Fig. 2 illustrates the design research process that we expect to use as we embark on this effort. Even though we will not be specifically developing and implementing the curriculum at this point, the figure illustrates the iterative cycles of development we will progress through with our community of advisors.

One of the most important challenges we face in this process is ensuring that we innovate as part of this process. Most of the members in our constituent community will have been exposed to many of the same educational practices which we seek to replace. It is therefore important that we weigh the feedback and ideas we receive from outside disciplines and students more heavily. It is equally important that this process be recognized as an ongoing, continuous process.

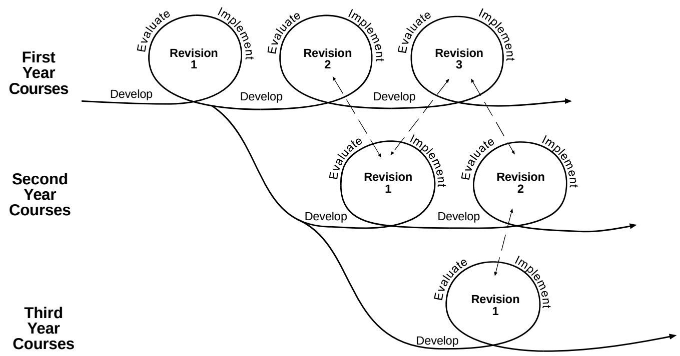

Figure 2: Illustration of the design research activity intertwined with the curriculum re-design. Each re-design cycle consists of development, implementation and evaluation. The fundamental learning helps inform future developments in program re-design.

2.3.2 Curriculum Threads

In our industrial advisory board meeting in May of 2006, they clearly defined several threads they would like to see throughout the curriculum that are introduced in the first year and then the program builds on these topics up through the senior year. The threads they defined included Multi-core programming, Graphics and simulation, Software Engineering fundamentals, Open Source Development and Professional skills, particularly oral and written communication. The purpose of the curriculum threads are to help provide continuity and progression throughout our courses so that students understand how topics are interconnected. In our ECE program, much of that continuity was provided through the Tekbot program [9], a technology platform which students continuously evolve and modify throughout their undergraduate career. In CS, a more abstract and diverse program, we plan to use the community of code to help re-enforce the curriculum threads.

The idea is not to force all threads into every course, but instead focus on the thread or threads which make most sense for each year in the program. The threads help establish a clearer progression between courses, as courses will be grouped according to the topics discussed, tools used, or projects given. By establishing and re-enforcing to these five threads when possible, we also lower the learning costs for students. For example, learning to understand the structure of Linux is a major hurdle if it is only used in a single operating systems class. The investment becomes more reasonable if the same system, Linux, is used as a case-study and source for examples in a software engineering class as well, or as a host system for a class on networking, computer security, or encryption. The five threads were emphasized by our constituents who not only hire most of our graduates but many of them were also graduates of our program at some point.

Multi-Core Programming

One of the key paradigm shifts facing computing professionals and students is the proliferation of multi-core technology, and the changes this represents for how programs are designed and written. Multi-core technology is now all around us. The latest generation of processors from all major computer manufacturers are now based on multi-core design, even modern graphic cards feature multiple cores.

This is one of the most important challenges facing the computer industry today. According to our industry partners, tremendous resources are currently being spent retraining programmers to deal with the kinds of concerns and programming approaches required in this new programming paradigm, once an esoteric area of study for undergraduates, only applicable for those interested in high-performance computing.

Our challenge therefore is to adapt to this new computing reality, rethinking and adopting all or most of our course and project work to train students to think and work with multiple cores, threads, and resource management from the beginning of their studies.

Graphics and Simulation

Graphics is pervasive in many computer systems today, most notably in computer games and animated movies. An added feature of emphasizing this technology is that students are motivated by the visual satisfaction they get when they make an application of this type. Computer graphics and simulation are also areas which appeal to different populations, improving our chances for attracting a more diverse student population. We are dealing with a visual-based generation; students who have grown up with digital images, who have grown up on Pixar movies and computer games. By allowing them into the “secret club” of those who know how to use their creativity and computer skills to create images and simulations, they will be able to combine their CS skills with a creative outlet that no other CS area is able to provide.

Images or simulations created by students can then be shared and demonstrated through the Community of Code described below, in a virtual gallery. Here students can post images that they have created, as well as discussions of how these were created, and share tips and tricks. The point would be to integrate exciting and useful graphics into the curriculum, and then give the students an outlet to brag about what they have done, and give them motivation to learn more and do it better.

Software Engineering Fundamentals

One of the most persistent comments we receive from our industry partners is the desire for stronger fundamentals in terms of software engineering and real-world project experience. This is a general comment that arises year after year from our advisory board. The feeling is that students often do not get enough exposure to real-world tools and experience with real-world tasks, meaning they are often ill-prepared to make the transition to the work-place.

One of the cornerstones of the platform for learning is hands-on learning throughout the curriculum. As part of this initiative, we will re-examine how and when software engineering principles and techniques are introduced and emphasized, seeking to give students the knowledge they need, when they need it in order to make the most of their project work.

This is an especially important thread because it directly affects incoming students’ ability to integrate and start to actively participate in the Community of Code described later in this proposal. In order to become productive members of the student community, students need to learn and adopt basic software engineering practices and learn how to use basic tools.

Through the software engineering process, we can help ease students into the *Community of Code*. As an example, though many first-term students are ill-equipped to join complex open source projects as programmers, they can be taught basic testing techniques which will help them make a contribution, become part of the community, and develop a deeper understanding of how software works. As students mature and are ready to start and manage projects of their own, they should be taught the project management skills and concepts needed to manage a team, manage a project. This is the kind of just-in-time learning, weaving through the curriculum, and especially targeted at helping students integrate and grow into the Community of Code that we seek to institute.

Open Source Development

At its heart, open source development simply means the production or utilization of software that has been created or released under various different forms of license or restrictions. Generally these licenses insist that the code for the application remain available for public inspection, and that any subsequent products or enhancements for such an application also retain these restrictions. But this simple description of the mechanics of open source does nothing to capture the culture, or mindset of the open source movement, or how this mindset has produced a fundamental shift in the software development process. In a thumbnail, the open source culture rejects the traditional centralized models of software development in favor of more loosely connected communities of interest, working together to product artifacts or products.

David Patterson, recent president of the Association for Computing Machinery, listed a course in Open Source as the number one technological change for computer science in the 21st Century, and as his number one "course I would love to take" [15]. Oregon State University, like most colleges and universities has slowly been starting to react to the open source movement. At Oregon State University we have a number of undergraduate students who participate in open source projects informally, and a smaller number work in a more formal setting under the auspices of the Open Source Lab (OSL). This fall the Open Source Educational Lab (OSEL) was created with the simple mission to introduce as many students as possible to open source techniques and ideas. By working with the OSEL, the department will seek to develop this sense of community and extend it to a wider group of students.

2.4 Building a Community of Learners – A Community of Code

Among the cornerstones of our *Platform for Learning* is the need to put lessons and activities in context, to provide opportunities for hand-on learning, to make them fun and enjoyable, and to promote ownership in order to encourage initiative, leadership, innovation, creativity, collaboration, and interdisciplinary work. The *Community of Code* brings the student community together, encourages students to pursue their ideas and passions, and ties them into the larger university community, who serve as clients and sources of ideas, inspiration, and guidance.

The Community of Code, as we envision it, is a combination open source repository and social networking site. The repository aspect allows and encourages students to engage in constructionist learning, learning by working on projects which are personally meaningful to them, which they take personal pride and interest, motivating their exploration and learning activities [4, 5, 8]. A key element to the constructionist approach is the idea of an audience of peers, and the importance of positive feedback and encouragement derived from working with others and showing off your achievements. Studies have shown that these types of environments can be very powerful when balanced correctly, encouraging women and minorities to participate [5]. The repository can also serve to allow students to pursue their projects across classes, as they seek out the knowledge needed to address specific needs they develop as part of perusing their ideas and dreams. As such, this could then facilitate more project-based learning in the classes.

The code repository, modeled on an Open Source repository, needs to be built in such a way as to suppress some of the less desirable social dynamics evident in some Open Source communities. Being designed for an educational audience, an audience with a great range of skills and abilities, not just between freshmen and seniors, but also within these different classes, it is essential to help everyone find a place. Open Source communities are often pure meritocracies [16], where technical skill is rewarded with status and power, and those perceived less skilled are shunned and ignored. Indeed, some of these dynamics are thought to account for the extremely low participation of women in Open Source (less than 1% of participants [14], even when compared to the rest of our discipline. Obviously, this is an unacceptable dynamic in an educational setting.

Our repository needs therefore to be built on a different reward structure and community model, where not only technical skill is valued, but soft skills such as communication, leadership, and writing ability, and mentorship. Attaining the right balance of incentives and social structures is an interesting challenge, and could open the way for more widespread adoption of the open source model in educational environments.

Community building within the student population is a key element to the success and integration of minorities, who can often feel isolated and alone. The second element of the

Community of Code is therefore a set of features to help students build personal connections. By ensuring they have access to a larger pool of fellow students rather than their classmates alone, we increase their chance of making that social connection, or identifying a role-model they can relate to, or giving them that broader perspective on computing which can be so crucial to their success.

The desire for a purely social component to the Community of Code idea is in part based on the success of social networking sites, like MySpace, with students, and in part in recognition of its critical role in helping break the ice which many new students, minorities and women in particular, may encounter when seeking to join student projects. It is always easier to approach someone who is not a complete stranger for help and guidance. Our idea is not to try and supplant sites such as MySpace, but to provide a safe, controlled, and limited community portal.

The community site will also serve as this communities window into the outside world, and how computing has come to be part of almost every discipline and part of life. Our intention is to invite the rest of the campus community and select partners to be a part of a limited side of the community, offering students sample projects, ideas, and mentorship. A job bank designed to encourage our students to seek out and join interdisciplinary teams and development efforts in order to better prepare them for the challenges they will face in their professional lives. This aspect of the community is inspired by the success of our year-long senior design course, where students work for a client, often someone outside the department, to develop a concept and system which will meet that clients needs. Students not only gain a tremendous amount of experience in the process, they also learn how to work with non-computer scientists, and to see computing in a larger context.

Tying the outside world to our student community also has the potential for introducing projects to the community which may be more appealing to women and minorities, often less interested in technology for technology's sake. In the OSEL efforts described earlier, a group of students have gotten involved with the One Laptop Per Child (http://www.olpc.org) effort, porting and refining the main text editing package which will be included in the system package. Projects such as these have the potential for attracting the attention and the imagination of a wider set of students, some because of the technical challenges, and some because of the social implications.

Finally, the Community of Code serves as a repository of work for us to use when communicating with prospective students about what computing is, and what opportunities it makes available. As will be discussed later, many of our outreach efforts have traditionally failed because of the extremely narrow view many prospective students have of what computer science is. By allowing this community access to some of the students' projects, we will e better able to communicate the diversity and opportunities for interdisciplinary work which will allow us to improve our enrollment and diversity figures.

In order to help us build and support this community we have enlisted the help of the Oregon State University Open Source Lab (OSL, http://www.osuosl.org). The OSL is one of the world's largest open source hosts in the world, hosting and providing community infrastructure to projects such as the Linux Kernel, Debian Linux, Gentoo Linux, GNOME, KDE, and serving as primary mirrors for the Apache foundation, the Mozilla foundation, including the Firefox browser and Mozdev extension development community. The OSL will lend their hosting facilities and their considerable expertise in building, promoting, and managing these types of communities. This combined with the considerable IT resources of the department itself should free the PI's from the day-to-day operation of community and infrastructure, and allow them to focus on developing and evaluating the features which make this environment unique.

2.5 Outreach and Dissemination

One of the overarching goals for this whole project, and indeed one of the reasons why this

NSF program was created, was to address the recruitment, retention, and diversity problems we face in this field. In order to bring the project full circle, it is therefore necessary to develop and implement an effective outreach program so we can reach potential students.

We find ourselves in a somewhat privileged position in this case, we believe, because the College of Engineering (COE), of which the School of EECS is a part, has already put in place an aggressive, and highly effective outreach program called the Ambassador Program (http://engr.oregonstate.edu/students/ambassadors/). The Ambassador Program is a marketing and recruitment program intended to combat stereotypes and misconceptions about careers in engineering. The program is a *student-to-student* recruitment/education effort using "near peers", engineering students just one to four years out of high school.

COE Ambassadors visit K12 schools throughout the state, and under the guidance of faculty and staff, develop engaging presentations to introduce engineering to pre-college students. Presentations stress concrete examples and personal stories of how engineering helps people and benefits society, aspects of a career choice that appeal to young women and minorities. The COE Ambassador Team is at least 50% female and 20% underrepresented minorities, reflecting the demographics of Oregon and the diversity goals of the college. The main goal of the school visits are to present engineering as a career option to students who may not have otherwise considered it, and to educate teachers and career counselors about opportunities for students in engineering. Each visit includes a stop at the school career guidance office and the distribution of engineering informational materials.

Now in its second year, the COE Ambassador Program has made engineering presentations at over 450 classes in 110 high schools, has hosted two "Introduction to Engineering" workshops for 87 high school career counselors, has participated in 27 career fairs/family science nights, and has directly impacted over 12,500 K12 students from all geographic areas of the state. As a result, the College of Engineering has had a 20% increase in incoming students fall 2006 as compared to fall 2005, and a 40% increase in the number of incoming freshman female engineers fall 2006 compared to fall 2005.

Unfortunately, this increase in female and underrepresented groups was not uniform across all engineering disciplines. While Bio Engineering, Chemical Engineering and Environmental Engineering have climbed to more than 40% of incoming students being female, only 4.12% of incoming Computer Science undergraduates are women. Despite these results we believe the basic plan is sound, we failed because we did not provide as tangible and meaningful examples of compelling and engaging student projects. Our challenge going forward therefore is to see if we can "catch up" by being better participants in this program. By leveraging the work done within the Community of Code, we can potentially tell a much more compelling story.

We will also leverage the Community of Code outside this context, making parts of it available as a public showcase and portfolio of student activities and achievement, both as a sample of what is possible when working in our discipline, as well as for demonstrating the value of the Platform for Learning approach.

There are three primary groups that we intend to leverage as part of our outreach efforts: the Math Task Force, the Computer Science Task Force and NCWIT.

The Math Task Force description

During the 1990s, college-level mathematics courses were individual with individual campus, resulting in much difficulty for students to transfer among the various colleges, community colleges, colleges and universities. The Oregon University System engaged a task force of faculty from across these multitude of institutions of higher education in Oregon, adding high school teacher representation with concern for courses taught at that level with the expectation of proceeding in mathematics as students enter college. Niess was a member of this two-year task force that reviewed, revised, reframe, renamed and redesigned the mathematics undergraduate curriculum that has since served students as they move among the various systems.

Computer Science Task Force

Since 1997, the Oregon University System partnered with the private sector through the Engineering and Technology Industry Council (ETIC, http://www.oregonetic.org/) with a concern that Oregon needed to develop a strategic resource for fueling Oregon's high tech and other sectors that use technology, including agriculture, healthcare, forestry, electronics, and retail. the number of engineers and computer scientists needed for the expanding economy. The partnership was made up of executives as well as leadership from Oregon colleges and universities. ETIC's work is designed to meet statewide goals to "grow our own" engineers and computer scientists. In 2005, ETIC's Computer Science Task Force (http://www.oregonetic.org/meetings/CStaskforce/index.htm) began efforts to expand the computer science pipeline in Oregon, making high school outreach one of their top agenda items in a broad-based effort to increase the number of Oregon students pursuing computer science education. Niess was added to the Task Force in recognition of the significant progress made in working with secondary teacher in integrating teaching about various technologies while simultaneously teaching science or mathematics. The assumption is that such an integration better academically prepares secondary student to pursue science technology, engineering and mathematics (STEM) careers – in particular computer science and software engineering. The next step in the work of the Task Force is to align the efforts at the secondary with the college level curriculum, particularly in computer science.

The various efforts in Oregon over the past ten years demonstrate the readiness for the ideas in this proposal and also a mechanism for leveraging the efforts funded through this grant.

3. Project Plan and Timeline

The key to the success of this effort is the collaborative team that has already been working together to tackle these issues. The initial project team includes the following:

Carlos Jensen serves as the project leader to ensure the goals and objectives of the program are met. Carlos provides research expertise in human-computer interaction, open source development, and the social dynamics of online communities with his passion for helping and addressing the needs of under-represented groups.

Terri Fiez supports the project by driving the innovations needed to transform the computer science curriculum. As a researcher, teacher and department head, she is passionate about creating a learning environment that welcomes a creative diverse student population.

Paul Paulson is a dedicated instructor of Computer Science. He has been an innovator in the program by deploying new instructional techniques and experiences. He has spent the last two years understanding how to incorporate open source development strategies into the classroom. He also has piloted efforts to understand effective approaches to platforms for learning in computer science.

Tim Budd is a dedicated educator with more than 13 textbooks used worldwide. He is passionate about the opportunities through involvement in open source and is a proponent of active learning in the classroom.

Mike Bailey brings an enthusiasm for educating the next generation of computer scientist along with his commitment to immersing students in a graphics and visualization environment. He has transformed the senior design experience at Oregon State University to where students work on customer driven projects with real deadlines and deliverables. Industry feedback has been extremely positive.

Maggie Niess is an education researcher who has collaborated on numerous research and education projects with faculty in the College of Engineering. She brings extensive experience both implementing, and evaluating curriculum change and learning. As an outside member of the project team she will provide a more objective perspective on evaluation and results.

Mike Quinn brings an experience base in addressing the programming needs of multicore processing and a strong commitment to undergraduate education. He has published a number of textbooks, most notably his recent text on computer ethics.

Our project plan, as outlined in the table below, involves two full iterations of the design process. Our revision A of the Platform for Learning are the plans which we have discussed in this proposal, and are included in order to give the Community of Advisors a running start rather than waste time evaluating something which we already view as being broken. In order to support the level of activity and urgency which we feel, our project is planned for 2 years rather than 3. This allows us to concentrate our resources in order to carry out a more aggressive and intensive program. Our goal is to reach a point by the end of this grant period where we can submit a proposal for a transformational grant.

Participant (s)	**Activities**	**Goals**	**Start**	**Completion**
Project Team	Implement Revision A plans for Platform for Learning, as described in proposal	• Implement designs based on current data, enable students and community of advisors to get a running start	Now	Summer 07
Community of advisors	Review of Revision A implementation, meet quarterly to evaluate progress, issue recommendations and ideas	• Provide feedback to Project Team	Fall 07	Spring 08
Student Team	Jump-start community of Code, mentor incoming students, gather data and observations for Community of Advisors, implement running changes and adjustments	• Provide prelim data for analysis • Support roll-out of community of code	Fall 07	Spring 08
Project Team	Evaluate and implement changes to Platform for Learning based on continuous student feedback and guidance from Community of Advisors. Meet weekly to manage and guide Platform for Learning,	• Establish platform for learning approach • Deal with ramp-up problems • Mentor students • Design and test evaluation Plan	Fall 07	Spring 08
Project Team Student Team	Finalize and Implement Revision B plans based on feedback from students and Community of Advisors	• Finalize Revision B plans • Implement Revision B of Platform	Summer 08	Fall 08
Community of advisors	Review of Revision b implementation, meet quarterly to evaluate progress, issue recommendations and ideas	• Provide feedback to Project Team	Fall 08	Spring 09
Student Team	Mentor incoming	• Provide evaluation	Fall 08	Spring 09

	students, execute evaluation plan, gather data and observations for Community of Advisors, implement running changes and adjustments	data for analysis		
Project Team	Evaluate and implement changes to Platform for Learning based on continuous student feedback and guidance from Community of Advisors. Meet weekly to manage and guide Platform for Learning,	• Mentor students • Perform evaluation	Fall 08	Spring 09
Community of advisors	Deliver final set of recommendations and evaluation result	• Issue Final report	Spring 08	Summer 09
Project Team	Perform final analysis of evaluation, design revised platform for learning, revised evaluation plan	• Issue Final report • Finalize revised Platform for Learning plan • Finalize evaluation plan	Spring 09	Summer 09

Evaluation is an integral part of this effort. The evaluation plan will involve first conducting a pilot project to assure that the work is on the right path. Niess, who has been the evaluator for the complete redesign of the upper level physics curriculum at OSU (funded by NSF, 1997-2000), has connections with all the above activities and will be able to plan a meaningful evaluation that will result in guiding the directions for the goal of promoting efficient and seamless transfer among the various education sectors. The pilot plan will be focused toward the development of a holistic approach to evaluation that considers more than a quantitative analysis of learning outcomes. The plan is designed to gather a breadth of evidence from the broader audiences with particular attention to the concerns for diversity and retention. The plan will engage community building and collaboration for gathering evidence from those various perspectives in order to provide information that is essential in the design of a curriculum that encourages and supports the diverse population that is needed in the future, that is concerned with providing evidence that can aid in the determining of resources and other needs in order to retain students in computer science career paths.

Summary

With the ensuing crisis of high technology jobs moving offshore, it is more critical than ever that engineering programs fundamentally change what is taught and the way it is taught. Engineering education has typically relied on traditional approaches to teaching (i.e., lecturing) the material because of the extensive depth and breadth of the content. While this instructional and curricular approach has been effective in the past, it does not address the needs of the diverse and creative workforce of the next decade and beyond. Our approach uses *learner centered experiences where the instruction and curricula* actively empower and engage the student as part of a learning community. Our track record of dissemination demonstrates that future developments will have a broad impact.