Student Engagement and Industry Readiness in a Systems Exploration, Engineering, and Design Laboratory (SEED Lab)

Dr. Vibhuti Dave, Colorado School of Mines

Dr. Vibhuti Dave is a Teaching Professor in the department of Electrical Engineering at Colorado School of Mines since 2011. She also serves as the assistant department head. She is heavily involved with undergraduate curriculum updates, assessment of learning outcomes and teaching core EE classes. Prior to Mines, she was at Penn State Erie, The Behrend College as an Assistant Professor in the Electrical, Computer, and Software Engineering program for 4 years. She received her undergraduate engineering degree in the field of Electronics and Communication from Nirma Institute of Technology, India in 2000. She received her M.S. in Electrical Engineering and Ph.D. (2007) in Computer Engineering from the Illinois Institute of Technology, Chicago, IL. Dr. Dave's research interests lie in the field of High Speed Computer Arithmetic and Computer Architecture. Her research has been focused on the design high-speed multi-operand adders. In addition, she is also interested in performing research in VLSI implementation of signal processing algorithms, and low power integrated circuit design. Her teaching interests include Digital Logic Design, Computer Architecture, Computer Arithmetic, VLSI Design.

Dr. Tyrone L. Vincent, Colorado School of Mines

Tyrone L. Vincent received the B.S. degree in electrical engineering from the University of Arizona, Tucson, in 1992, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 1994 and 1997, respectively. He is currently a Professor in the Department of Electrical Engineering and Computer Science, at the Colorado School of Mines, Golden. His research interests include system identification, estimation, and fault detection with applications in materials processing, and energy systems.

Dr. Megan Sanders, Colorado School of Mines

Megan Sanders is the Senior Assessment Associate at the Trefny Innovative Instruction Center at the Colorado School of Mines. Before joining Mines, Megan worked at the Eberly Center for Teaching Excellence and Instructional Innovation at Carnegie Mellon University, where her role focused on supporting instructors in conducting research about student outcomes in their courses. Megan's disciplinary background is in educational psychology. She earned her PhD from the Ohio State University, and her research focused on the idea of relevance in higher education—how we define it, how students perceive it, and how to measure it—an interest that continues to inform her work.

Dr. Stephanie Claussen, Colorado School of Mines

Stephanie Claussen's experience spans both engineering and education research. She obtained her B.S. in Electrical Engineering from the Massachusetts Institute of Technology in 2005. Her Ph.D. work at Stanford University focused on optoelectronics, and she continues that work in her position at the Colorado School of Mines, primarily with the involvement of undergraduate researchers. In her role as an Associate Teaching Professor, she is primarily tasked with the education of undergraduate engineers. In her courses, she employs active learning techniques and project-based learning. Her previous education research, also at Stanford, focused on the role of cultural capital in science education. Her current interests include engineering students' development of social responsibility and the impact of students' backgrounds in their formation as engineers.

Student Engagement and Industry Readiness in a Systems Exploration, Engineering, and Design Laboratory (SEED Lab)

Introduction

Laboratory courses have been a key component of engineering education in the United States since the founding of the earliest American engineering schools [1]. Today, well-designed laboratories in the undergraduate curriculum play a critical role in the development of students' hands-on skills, problem-solving abilities, teamwork skills and analytical thinking while also deepening the content learned in lecture-based classes. The primary focus of this paper is the design, evaluation and improvement of a multi-faceted, intra-disciplinary laboratory called the Systems Exploration, Engineering, and Design Laboratory (SEED Lab). Created with the support and input of industry partners, the SEED Lab aims to emulate our students' likely future experiences in a professional environment. The course employs assessment techniques such as reflection logs, CATME evaluations, team presentations at regular intervals, performance-based demonstrations, and case studies.

This work will contribute to the body of engineering education scholarship by evaluating the impact of the SEED Lab on students' development of attitudes toward failure in the engineering design process and reflection abilities. While this course was originally developed for use in an electrical engineering program, its key components, the structure of the projects, the desired learning objectives, and the course processes could be adapted for the benefit of engineering students in all disciplines.

Background

The vision for the SEED Lab was to provide a laboratory experience in which students develop key professional skills [2]. While core engineering courses teach engineering science theory, graduates experience a very different environment upon entering the workforce where they are presented with complex projects that require multi-disciplinary teams to complete. Some of the professional skills required in the industry setting include being able to work in teams, to prototype and experiment, to troubleshoot and fix problems, to recover from failure, and to reflect back on previous work to improve future performance.

Project-based learning and inductive learning methods have been shown to be effective pedagogical methods to target such skills and provide a motivating experience for students [3], [4], [5], [6]. Several universities across the country have moved away from a traditional lab environment by restructuring and redesigning their laboratory courses to incorporate project-based learning and teamwork. Examples of such efforts include an interdisciplinary course in semiconductor processing [7], a class focused on the design of a solar racing car [8], and embedded systems courses developed for a range of student levels [9], [10], [11], [12], [13].

From prior work, some common recommendations for the effective implementation of project-based learning emerge. Project-based learning helps engage students in the learning process and gives students a realistic, hands-on experience. It is also beneficial to incorporate collaborative learning methods by allowing students to work in cross-functional teams where each student brings a different skill set. Finally, settings that mimic industry practices help to prepare students to join the workforce after they graduate.

The SEED Lab incorporates these recommendations into the design of the course. This course is a required course for all EE majors and is taken by students in their second semester of junior year or in the first semester of their senior year. Students are expected to have a background in feedback control systems and embedded systems. Students take on roles that require developing differing knowledge bases from their teammates. Furthermore, because communication is required both within teams and between teams, students encounter the communication challenges inherent in multi-disciplinary teams, even though they share the same major. In addition to this, we have structured the course to enhance students' abilities to experiment and prototype as a team in the engineering design process and improve students' use of reflection as it pertains to solving complex engineering design problems. This vision for the SEED lab is also compelling to industrial partners that hire our graduates, and has led to significant financial support from a local company.

The elements that characterize the SEED Lab are the following:

- Each semester, the SEED Lab is organized around a single challenge problem whose solution requires the application of material from multiple courses in reflection of the complexities of the real-world. The solution of the challenge problem requires solutions of intermediate, domain-specific problems, as well as the integration and debugging of these smaller systems.
- In the first three weeks, students choose a specialization and complete exercises that develop their knowledge in that specialization. Examples of specialization include computer vision, control, processor communication, and sensors. Students will later form teams that include one member from each specialty. This provides a model for multi-disciplinary teams where the background of each member is different. Because all students are electrical engineers, we call this an intra-disciplinary team.
- In the next two weeks, students form groups of four and execute a small mini-project. This provides a format to introduce best practices for team work and establish team operation (including a team contract).
- In the remainder of the course students work on the larger challenge problem. Three demonstration periods are scheduled where teams demonstrate the current capabilities of their solution. Part of the grade is assigned based on the relative performance of the team’s solution compared to the best team. Because performance is measured in several categories, the team that is best in one category may not be in another. The demonstration periods provide key points for groups to observe and reflect.

Our interest in experimentation and prototyping has been motivated both by our anecdotal observations of many senior design teams and advice from industrial contacts. In the senior design process, students often attempt to build a final product without adequately planning for subsystem interactions, non-ideal component behavior, verification of engineering assumptions, and other aspects where prototyping and design iterations are useful. In addition, we observe that students are often tentative about experimenting without a well-defined solution in sight. A Stanford University study [14] originally designed to characterize and categorize common prototyping techniques adopted by students also reported student reflections toward prototyping. The study found that students saw prototyping activities as a place to start, a way to gain insight into their design, and a means to improve their work practice efficiency, motivating SEED Lab's push toward repeated prototyping cycles.

According to Kolb's Experiential Learning Theory [15] [3], reflective thinking is a key skill required for knowledge construction and life-long learning. There are increasingly frequent calls to incorporate reflection into engineering education. The motivation to incorporate reflection into higher education coursework stems from pedagogical research suggesting that students need to put meaning to experience to become successful professionals. As Dixon explains, “Experience per se does nothing for designers – or anyone else – unless they learn something cognitive in the process” [16]. Adams *et al*. specifically focus on reflection on experience as an intentional thinking process [17]. They point out that, in order for reflection to have any impact on future action, the individual has to revisit their past, and use one or more lenses to assign meaning to the experience.

The current version of SEED Lab incorporates weekly or bi-weekly student reflection logs. We use insights gained from [17] and [18] to use reflection as a learning tool for students, foster the idea that failure is not a bad thing, and as a tool to assess the research goals of this work. In [19], the authors have successfully incorporated reflection essays into computer engineering courses in which the authors provide specific reflection prompts that they used in their project-based junior level class. We have modified these prompts to better suit the needs of SEED Lab with the purpose of giving students an opportunity to reflect on two important questions: “What did I learn from my experience (good or bad)?" and, “How can I use the lessons learned from my experience in the future?". The prompts obligate students to focus specifically on things that did not go right and the techniques they developed to fix the problem. This could range from debugging strategies that they came up to solve a technical issue to nipping a faulty implementation strategy in the bud. We are using reflection logs as a means to track changes in students’ attitudes toward failure and view failure as an important step to success.

Course Design and Delivery

In this section, we describe the concept and operation of the SEED lab. When a laboratory is dedicated to a single course, the target applications are simplified so that students only focus on the details applicable to that particular course. However, real life applications are often complex, with several systems that must work in tandem. We have attempted to structure a laboratory experience so that students can obtain experience at a deep technical level in a particular area, while at the same time working on applications that require multiple complex systems to work together. As part of this process, students develop and utilize important professional skills, defined by ABET outcomes a – f [2]. This laboratory is organized around an overall challenge problem whose solution reflects the application of material from multiple courses and better conveys the complexities of the real world.

- The course was first implemented in Fall of 2015, and we are currently in the third year of implementation. Each academic year, a new challenge problem is posed. The challenge problems for each year have been:

- AY15/16: Your team is assigned by a company to design and build a prototype of a two-wheeled balancing robot that can carry cargo to a desired destination, which is expected to have a viable market in hospitals and other institutional settings.
- AY16/17: Your employer would like to develop an automated warehouse with small robots that will carry goods from where they are stored to where they are packed and shipped. Your team is assigned to design and build a prototype device. This prototype must be able to following the directions around the warehouse, which are given by signs. The signs include left arrows, right arrows, and stop signs. The robot should be able to detect a sign, travel towards it, and then follow the directions (turn left, turn right, or stop).
- AY17/18: Your employer would like to develop an automated warehouse with small robots that will carry goods from where they are stored to where they are packed and shipped. Your team is assigned to design and build a prototype device. This prototype must be able to determine its current position and orientation using lighted beacons that are located at fixed, known locations along the sides of the warehouse, and then proceed to desired locations within the warehouse as specified by coordinates.

The solution of the challenge problem requires solutions of intermediate, domain-specific problems, as well as the integration and debugging of these smaller systems. At the beginning of the lab, each student chooses one of the disciplines to concentrate on. In Figure 1, this is illustrated for the self-localizing robots assigned in AY17/18. Students choose to become experts in one of the four specialty areas. Students with a solid foundation in feedback controls typically choose simulation or localization. Students interested in working on embedded systems with strong programming skills choose processor communication or computer vision. As they work through the course, students start in discipline-specific groups to solve intermediate problems, and then move to their systems groups for integration.

For the first three-weeks, students spend time becoming experts in their chosen area of specialty. These discipline-specific groups are small, numbering one or two students. In this phase, students are given introductory handouts exposing them to hardware and software they are expected to use to build the final product. For the self-localizing robot, these include Arduinos (for motion control implementation), Raspberry Pis (for computer vision), the I2C protocol (for processor communication) and MATLAB simulation (for control design). Students typically entering this course have little to no experience with these tools. The introductory handouts point them to resources that they can use to learn the tools and mechanisms. As a form of assessment of their learning, they are also asked to demonstrate some simple hands-on exercises at the end of this phase. One of the main reasons for using Arduinos and Raspberry Pis is the plethora of resources available to help one learn how to interface with them. Handouts are specifically designed to allow the students to delve in to these resources and pick up information as needed. Lectures and step-by-step guides are avoided unless absolutely required. For example, we do provide a tutorial on how to remotely connect to a Pi, but they must write their own code to communicate between the Pi and the Arduino using on-line resources to learn about the I2C protocol. The short "do-it-on-your-own" exercises give students the opportunity to assess their own understanding. The requirement that they demonstrate these exercises successfully ensures that when they become a part of a team, they are well-prepared and ready to start with the task at hand.

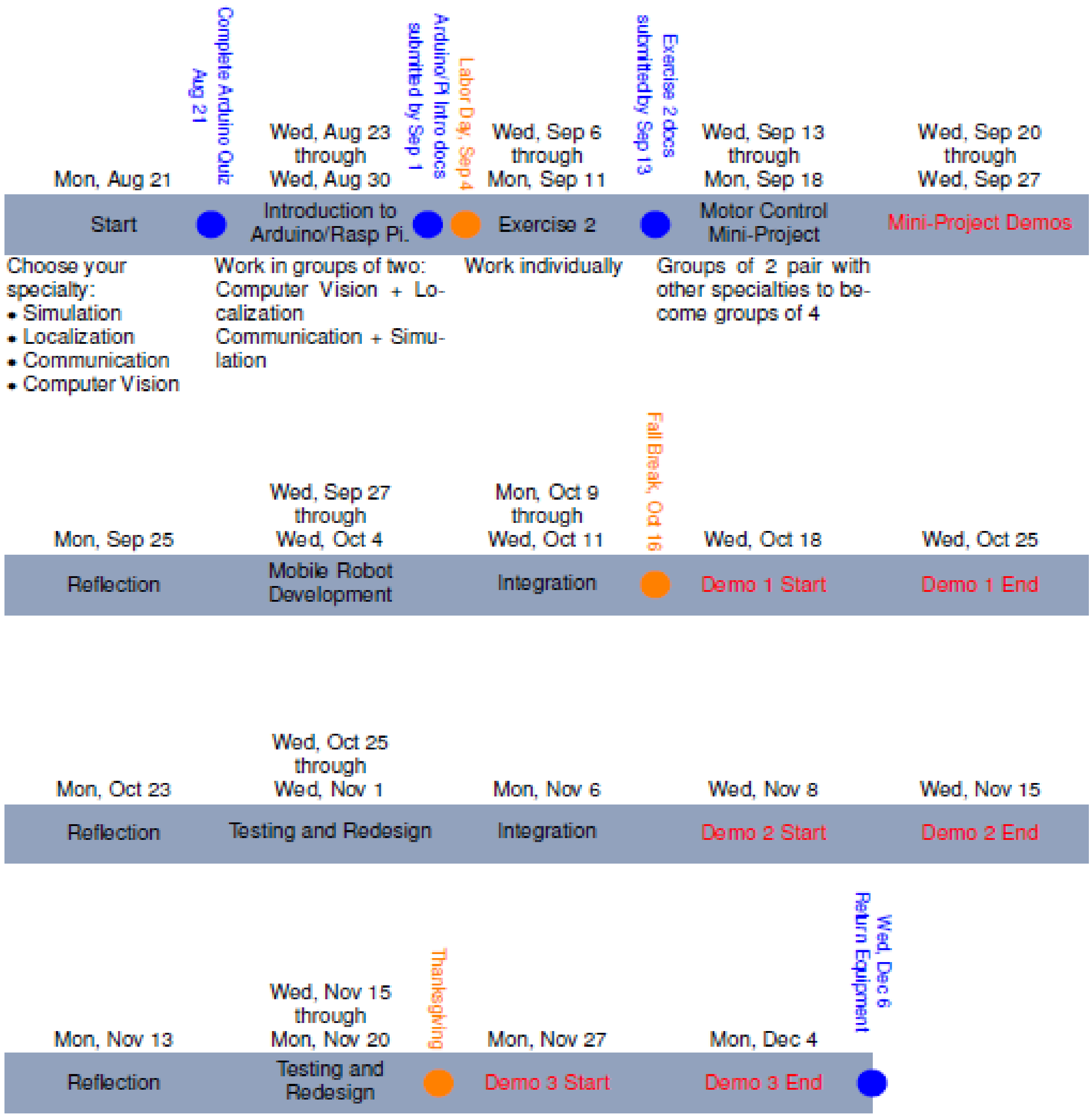

Figure 1: Semester Schedule of Activities for SEED lab

The next phase of the course brings together four students, one from each area of specialty, to work as a team on a mini-project. The mini-project is designed to incorporate the different disciplines as sub-systems. The students typically divvy up the work so that each student is working on a sub-system based on what they have learned in the past three weeks and finally requiring them to integrate the subsystems together. The mini-project acts as a warm-up activity for students before the main challenge begins. This activity gives students the opportunity to get to know their team, and get comfortable with the team dynamics. An example of the mini-problem statement from AY 17/18 is provided in Appendix A. The handout for the mini-project is more of a guide, rather than a procedural, cookbook-style lab handout, to ensure student success with proper effort and strategy. A section in the handout details what the different project elements are. The following are examples of the connecting pieces from the mini-project used in AY 17/18:

- OpenCV to implement color detection,
- Motor driver and Arduino to spin a motor,
- Arduino to read the rotary encoder,
- I2C protocol to set up communication between the Pi, the Arduino, and the LCD so that the distance measured by the Ultrasonic sensor can be displayed on the LCD,
- Simulation in MATLAB for the motor, and the mathematical design of a controller for the motor.

The third phase of the course is when the students start working on the overall design challenge, divided in to three cycles. At the end of each cycle is a demonstration. The requirements of each demo build on the previous one. For example,

- In demo 1, the robot needs to detect a beacon and calculate the angle required to rotate the robot to reach the beacon by moving forward. Another independent task is to rotate the robot by a specified angle, move forward and stop after a specified distance.
- In demo 2, the requirements get more rigorous. The robot must be able to detect two beacons, use these beacons to determine the starting location of the robot, and then move to a second specified location (relative to the beacons).
- The final demo demonstrates the full implementation of the robot. The robot is placed in a 10’ X 10’ grid, with two or three LED beacons placed on each side of the square grid. The robot is placed in a random position within the grid and is required to travel within one foot of each of the programmed positions in a sequence (four positions total). Between each demo, students can refine their design to make it operate better and rectify buggy behavior.

Course assignments

In order to assess student progress and competency, we use three forms of assessments.

1. Documentation: In the first phase of the course involving the introductory exercises, students are required to submit documentation after completing the exercises. This consists of well-commented and easily readable code along with a document answering basic questions pertaining to the exercises at hand.

   Starting with the mini-project, successful completion of the task in a systematic way is of importance. The documentation requirements are for students to lay out strategies used to complete the task. This is done in the form of a short presentation where they provide an overview of their design and talk about the problems they faced when completing the design, experiments they designed to debug the problems, and methods used to implement a solution. To evaluate teamwork, students are also required to submit team meeting minutes from every week, detailing specific tasks assigned to each member of the team, and overall goals for the week.

Finally, students fill out CATME surveys at the end of each design cycle to provide instructors with insights in to team dynamics. Instructors intervene if a need is indicated by the CATME results.

2. Demonstration: Although demonstration of the introductory exercises is required, the stakes get higher as the semester progresses and the students enter the design phase. At this point, the course takes the approach of performance-based scoring for the demonstration piece. Each team is evaluated with reference to the best-performing team. For each demo, there are several performance criteria provided to the students. For example, the performance metrics for the self-localizing robot are: number of failures to detect the beacon, average error in reported angle to beacon, average error between desired position and actual position and average time to complete the run. The team with the best score in a particular category sets the benchmark for that category. Teams can choose to focus and maximize their score in a certain category while trading off on the remaining categories. This adds an element of competition which serves as motivation for the students.

3. Reflection Logs: For the first phase, where students perform preliminary exercises, they fill out the reflection logs in three different segments. The exact prompts can be found in Appendix B. The expectation is for them to answer the first two prompts before they start working on any exercises. These pertain to what their goal is for the week, followed by what background information they are starting with. Students need to keep track of resources utilized and documentation created which they can refer back to in the future, especially since some of the methods and tools are brand new to them. As part of their reflection log, students are therefore asked to list all resources used to complete exercises as well as any kind of documentation created. This is best done while they work on the preliminary tasks. At the end of this initial phase, we want students to reflect on the difference in their knowledge and understanding from when they started to the completion of exercises. Responses to the last two prompts are meant to be completed at the end of the week. We ask the students to reflect back and provide an example of something they would do differently or could improve upon during the course of the exercise. The final prompt is to rate their comfort level with equipment/tools that they worked on during that week.

   The remaining three phases of the course are for students to build and deliver a final working product. During these phases, reflection logs focus on their experience and development as a team member. We also ask them to reflect back on their team dynamic so that the instructors can get a better insight in to the working relationship of members within a team. The logs are divided into three parts: a weekly overview of goals set for the team and tasks assigned to them, reflection on their own learning through any false starts they had and debugging mechanisms they developed to solve, and reflection on their team dynamics.

Required Resources

The challenge problem is related from one year to the next, which allows the re-use of a large portion of the equipment from year to year. We have focused on robotics applications initially. However, a laboratory course needs continuous updating as new processors, sensors, software, and other technology becomes available, and the development of new challenge problems every year supports this necessary work, as well as ensuring that teams do not rely on past students or material. The format of the course could applied to a wide variety of applications, and we are working on developing a power micro-grid application in the future.

Each student is issued an Arduino starter kit and a Raspberry Pi starter kit. These kits are used with the two preliminary exercises: an introduction to the development environment of each embedded system, and a specialization exercise that introduces some concepts and coding techniques useful within that specialization. The Arduino starter kit includes the Arduino, cables, LEDs, resistors, pushbuttons, breadboard and an ultrasonic sensor. The Pi starter kit includes the Pi and case, SD card, charger, camera board, small keyboard and mouse, LCD screen, and cables to connect the Pi to a monitor. Each team of four students is issued a hardware kit and a battery kit that is used (along with the parts from the preliminary kits) for the mini project and challenge project. The hardware kit includes motors, motor driver shields, Bluetooth shields, tools, mounts, screws and nuts, switches, fuses, ball-casters, Actobotics® channel and brackets of various sizes, and wheels. Spare parts for fragile items such as motors and motor drivers are included. The battery kit includes sealed lead acid and Li-ion batteries with chargers, with enough batteries available that spares can be charging while others are in use.

The total cost of the kits per student is $284. This cost excludes standard lab equipment such as oscilloscopes, DMMs, desktop PCs, storage lockers, etc. The kits need to be renewed each year due to broken or worn-out equipment or consumables. This cost is typically $20-$30 per student per year.

The laboratory course is overseen by two instructors and three undergraduate teaching assistants. The undergraduate teaching assistants attend three hours of regularly scheduled lab time per week, and are also available in the lab for two other hours per week for consultations. This staffing has been sufficient to oversee about 40 students per semester. Just as the teams specialize, so too do the instructors. Two instructors are useful because the broad range of subject matter and the need to support two different embedded processors and programming languages, along with the development of four different specialization activities. In our case, one instructor is more familiar with the Raspberry Pi, computer vision, and processor communication, while the other is more familiar with the Arduino, control and localization.

The final required resource is the development process that allows for continuous improvement of the course and modification of challenge problems. Initially the teaching assistants and students were supported over part of the summer. During this time, the teaching assistants worked through beta versions of assignments and projects, instructors developed new material, the challenge problem was outlined, and a prototype system was built to verify that a solution is possible using the equipment available for the lab, and achievable within the time available. Now that the course has been set up, we will be moving to a system where students who have already taken the course will be used as teaching assistants, and challenge problem development will be part of the work for the teaching assistants during the year.

Assessing the Impact of the Course on Student Learning

Although student work has provided information about the impact of the course on individual students in the past, we are currently engaged in a more formal assessment of the Spring 2018 offering of the course. Two data sources are being used to assess the change in students' attitudes and approaches to engineering design and students' abilities to reflect and learn from failure: a case study activity repeated at two time points over the semester and the reflection logs that students complete as part of their regular coursework.

Case study activity

First, a case study activity was administered at the beginning of the semester and will again be presented at the end of the semester. This activity was designed to elicit students' general approach to the design process, rather than a solution to the particular case (see Appendix C). The implementation has two parts. Students first complete the activity on their own for homework, submitting their responses individually. Then, the instructors debrief the activity during the next class to reinforce the themes that emerge across the class.

We piloted the case study activity at the end of the Fall 2017 offering of SEED Lab. This pilot was used to test the prompt students were asked to respond to and the implementation of the activity during class. The instructors made adjustments to the implementation of the activity in the Spring 2018 course based on the Fall 2017 pilot. In the Spring 2018 assessment, students' individual beginning- and end-of-semester responses to the case study activity will be scored using a rubric developed to evaluate design thinking. The rubric is based on the *Design Thinking Assessment Rubric* developed at Stanford University's Hasso Plattner Institute of Design [20]. Using this as a pre-test/post-test measure will help establish students' abilities upon entering the 2018 course and the growth in those abilities across the semester.

Student reflection logs

The second source of data that will be used to assess the impact of the course is student reflection logs. Although the reflection logs have been an important part of the course design since the beginning, the logs from Fall 2017 were piloted as a data source for the formal assessment of the course in Spring 2018. Students' reflection logs were coded using a general inductive approach [20]. In this approach, codes are not defined *a priori*, but instead, themes are allowed to emerge from the data. As such, this approach is similar to other qualitative analytical approaches like grounded theory [21] [22]. One of the principal investigators first read through the entire set of student reflection logs, then developed initial codes, and refined and reduced these codes through the coding process. Each response could be assigned multiple codes. Codes were then compared, contrasted, and related to one another to establish the more prevalent overarching themes in the data [20]. Themes are presented with representative examples.

Two sets of reflection logs were analyzed: one which was written relatively early in the course, and one which was the final set of reflection logs of the semester. The first was written at a point in the course where students have finished individually developing their skills in their respective areas of expertise and have started to work in teams toward a common goal. This set is referred to below as "Mini Project Reflection Log," as the students are beginning work on their mini project. The second set of reflection logs was written by students after completing the third demonstration, in the two weeks leading up to the final demonstration of their design. Though students wrote reflection logs for each of these two weeks, we combined them into a single set (called the "Final Demo Reflection Log") in an attempt to analyze a complete picture of this point in the design process.

It proved difficult to compare student responses to these two distinct reflection logs, because the responses on the first and second logs were generally very different, even when looking at the same student. Unexpectedly, the quality on the first log (the Mini Project Reflection Log) was generally much higher than that for the second, the Final Demo Reflection Log. We suspect that the reasons for this are two-fold: 1) the Final Demo one is written at the end of the semester, when students are overwhelmed with other assignments and preparing for finals (four students did not even complete these final reflection logs), and 2) by this point in the course, many of the teams had final designs that only required minor modifications each week, if any.

Using the qualitative data analysis approach designed above, we were able to draw the following conclusions, which are organized by their relevance to our two desired student outcomes:

Impact on students' attitudes toward and approaches to engineering design. Teamwork was a frequent theme in the student reflection logs, which was likely in part due to the fact that approximately half of the questions related to team performance and practices. Students often observed that the drawbacks of working in a team (challenges in finding a time to meet, resolving conflict when there were different ideas about how to approach a problem) were balanced by the benefits (multiple people to approach a problem, the ability to bounce ideas off each other, diverse strengths and abilities). When asked for evidence about how well their team was performing, some students demonstrated a less refined view of the role of teamwork by directly linking the performance of their team and the performance of their robot (for example, stating that they determined their team was high-performing because their robot had successfully completed the most recent demo).

Teamwork was also found to influence every part of the design process. The students often linked it to the success (or lack there-of) of integrating their sub-systems together into a unified final robot design. For example, one student wrote,

> We thought the project would be a simple integration of the individual subsystems, with little additional work, so we mostly worked independently. It wasn't until the [week before the completion date] that we realized the amount of additional work that was involved.

Students frequently cited the integration of the sub-systems as the hardest part of their work. For example, a student observed, “It [was] difficult to combine code because everyone’s code would work individually. However, when we would try and combine each individual part, it would occasionally not work and then you would have to look through both people’s code to try to determine why it wasn’t working.” Successful teams worked together on this integration and understood individually that they would need to develop a basic knowledge of the other sub-systems in order for such integration to work.

Team dynamics also were frequently linked to the debugging process. In high-functioning teams, the students admitted that having different ideas for debugging initially introduced challenges, but that this ultimately allowed them to debug faster or result in a better final design. Students who communicated a rudimentary approach to debugging often also possessed a simplistic view of team-based systems design. For example, one student wrote in the final weeks of the course that his team had no false starts in their design process because of sheer will: “We really do not want any extra work for this class so we are careful not to.” This is an attitude that likely bears little resemblance to the reality of design in an authentic industry environment. Such students also often wrote in the first-person when describing their debugging, which relays a dependence on their own individual abilities rather than the collective skills of their team.

Impact on their abilities to reflect and learn from failure. Another objective of the SEED Lab course was to develop students’ reflection abilities, especially as it pertains to engineering design and learning from failure. The reflection logs were also analyzed to determine the impact the course had in this area.

Unfortunately, the quality of the students’ reflections did not seem to improve over the course of the semester. In fact, a noticeable trend was that their reflection log responses were more succinct and less thoughtful by the end of the semester; however, this could be due to the hypothesis described at the start of this section (by the final week of the semester, the students may have been exhausted and overwhelmed by other academic commitments, as well as the fact that their designs were largely finalized with little to reflect on). Furthermore, not a single student referenced the role of reflection in the design process. This could be a product of the questions being asked by the reflection logs, something that we will discuss in the final section of this paper.

A challenge of this work is that what we are analyzing—the reflection logs—really measures the students’ abilities to articulate their design approaches, rather than the complexity of the approaches themselves. For example, when asked why he and his team pursued a false start, one student wrote, “The decision that led us to this false start was the decision to write incorrect code.” While this statement is likely true on its surface, it fails to show any insight the student had into their prototyping process. Thus, the following analysis should be interpreted with this in mind—that we are inevitably conflating students’ reflection abilities with the diverse design abilities we are targeting.

The students generally had a difficult time reflecting on their design process, including articulating their team goals. For example, when prompted for their goals for the week, a handful of students described simple and obvious objectives such as, “"Our goal [this week] was that each person would create high quality material," and, "Complete the mini project." Such goals likely did not lend themselves to useful project management. This is an example of where reflection *should* be an integral part of the design process, with straightforward yet vital questions such as, “What is my team doing this week? What am I contributing to this goal? What are my tasks as an individual?” Yet some students failed to complete even this basic reflection task in a meaningful way.

The reflection logs asked many questions about the debugging processes students used when they encountered false starts or challenges. The responses to these questions provide a lot of insight into their views on debugging and their abilities to debug complex problems. A handful of students had very self-focused views when discussing any false starts they encountered, frequently using first-person pronouns to describe their work. For example, rather than discuss any work done as a team, a student wrote on his final reflection, “This week I did not do programming. As a result, I have not encountered problems.” In writing this, he is failing to take ownership for any issues that *his team* ran into. In addition, some students went a step further, blaming failures on specific teammates and not taking ownership themselves. One student demonstrated this when he wrote, “[Student A] decided to change out the raspberry pi which made it so my code could not be tested. If he had not done that, the team would have had plenty of time to test the robot and make sure that it worked.”

When describing their debugging approaches, some students displayed a weak understanding of debugging. Their processes included simply rereading their code or comparing their product to one found online, to see where any errors were. While these are no doubt useful approaches, they are straightforward and convey a belief that “false starts” are merely due to simple mistakes which are obstacles to achieving a single “right” outcome, rather than necessary steps in the design process toward ultimately creating the best final product.

Approximately a third of the students in the course were able to clearly articulate their debugging approaches, often in a very detailed and insightful way. One student wrote,

> The main debugging technique that we used was to output values. We had numerous issues with getting the control system to work correctly. We tried isolating different pieces of code, commenting out conditionals, and checking values until we found our problems. In the case of the switch bouncing issue described above, we used variable output to deal with the problem. Although we were not able to find a perfect solution, by outputting the value of the go/stop line with each iteration, we were able to identify that our initial proposed solution was not effective.

Clearly, he was involved in this process and understood why his team undertook each step. While many students described debugging errors in their code, a few conveyed insights about the interplay between software and hardware, and detailed steps they followed to catch issues with their hardware. One team determined that their robot battery was not providing an adequate voltage; another learned their motor was not operating as expected. We propose that this is an important outcome of this course, as debugging software and hardware requires the students to apply their debugging skills to very different sub-systems. One student managed to summarize the lessons he learned about debugging in very clearly outlined points:

> The most important thing about debugging in this class is to 1. SLOW DOWN and think about what you're doing and 2. Use all debug tools available. The first one is absolutely crucial, as when working in group it can be tempting to just start trying stuff and see if it works, but systematic debugging is extremely important.

Finally, in some of the weaker reflection logs, students struggled to articulate both the false starts their team encountered and the debugging approaches they pursued to address such setbacks. This fact points to the possibility that such students have failed to develop both of the interrelated skills of reflection and debugging/overcoming false starts.

Conclusion and Future Work

Analysis of the reflection logs showed that they may not be exactly measuring what we want them to measure. For example, students rarely referenced the design process in their reflections or the role reflective practice can play in engineering design. In Spring 2018, this was addressed by including a component called "Reflection on Reflection". After students submit their reflection logs, they come together as a group to discuss the challenges they have faced and experiments that they set up to solve problems. This discussion occurs with the course instructors acting as moderators thereby assisting them in the reflection process. While getting students to explicitly discuss such concepts is undoubtedly difficult, changing the questions on the reflection log may encourage the students to make such connections. This may require different reflection logs for different points in the course, to target specific learning objectives throughout the semester.

In the future, a more comprehensive analysis of the student reflection logs throughout the semester could lend itself to understanding how student views and abilities shift with time. This was not possible with the two sets of reflection logs that we chose to look at. Such an effort may be more effective with refined reflection logs that more accurately assess our research objectives. Supplementing such analysis with other forms of assessment (for example, focus groups or analysis of student assignments) would also allow us to make more accurate research findings.

The research study being implemented during the Spring 2018 semester is designed to provide more targeted information about student learning throughout the course. Based on preliminary findings from the Fall 2017 pilot of the data collection methods, both the reflection logs and the case study were refined for the Spring 2018 implementation of the study. These changes were made to better align the reflection log prompts with the aspects of student design thinking we are interested in assessing. The changes to the case study activity were implemented to focus student attention on the design process in general, rather than a specific solution to the particular case study scenario. Thus, data collection underway during the Spring 2018 semester will help us better understand changes in student design thinking as a result of taking SEED Lab.

Other future work will focus on evaluating the longer-term impact of taking SEED Lab. For example, future work will examine student performance in subsequent design courses, comparing students who took SEED Lab with students who did not take SEED Lab. Additionally, the study being conducted in Spring 2018 will help establish a baseline assessment of the course's effectiveness. Having baseline data will allow the instructional team to empirically test the effectiveness of any future changes to the course's design and implementation. Taken together, the pilot data from Fall 2017, initial study in Spring 2018, and future work will help demonstrate the effectiveness and impact of SEED Lab.

References

[1] L. Feisel and A. Rosa, "The Role of the Laboratory in Undergraduate Engineering Education," *Journal of Engineering Education,* vol. 94, pp. 121-130, 2005.

[2] L. J. Shuman, M. Besterfield-Sacre and J. McGourty, "The ABET \professional skills" - can they be taught? can they be assessed?," *Journal of Engineering Education,* vol. 94, no. 1, pp. 41-55, 2005.

[3] M. Abdulwahed and Z. Nagy , "Applying Kolb's experiential learning cycle for laboratory," *Journal of engineering education,* vol. 98, no. 3, pp. 283-294, 2009.

[4] A. Hofstein and V. N. Lunetta, "The laboratory in science education: Foundations for the twenty-first century," *Science education,* vol. 88, no. 1, pp. 28-54, 2004.

[5] T. Moore and H. Diefes-Dux, "Developing model-eliciting activities for undergraduate students based on advanced engineering content," in *Frontiers in Education, 2004. FIE 2004. 34th Annual*, Savannah, GA, USA, 2004.

[6] M. Prince and R. M. Felder, "Inductive teaching and learning methods: Definitions, comparisons, and research bases," *Journal of engineering education,* vol. 95, no. 2, pp. 123-138, 2006.

[7] A. J. Muscat, E. L. Allen and E. D. Green, "Interdisciplinary teaching and learning in a semiconductor processing course," *Journal of Engineering Education,* vol. 87, no. 4, pp. 413-421, 1998.

[8] D. Carroll and P. Hirtz, "Teaching multi-disciplineary design: Solar car design," *Journal of Engineering Education,* vol. 91, no. 2, pp. 245-248, 2002.

[9] D. L. L. Maskell and P. J. Grabau, "A multidisciplinary cooperative problem-based learning approach to embedded systems design," *IEEE Transactions on Education,* vol. 41, no. 2, pp. 101-103, 1998.

[10] M. Barnes, Bailey M, P. Green and D. Foster, "Teaching embedded microprocessor systems by enquiry-based group learning," *International Journal of Electrical Engineering Education,* vol. 43, no. 1, pp. 1-14, 2006.

[11] L. Jing, Z. Cheng, J. Wang and Y. Zhou, "A spiral step-by-step educational method for cultivating competent embedded system engineers to meet industry demands," *IEEE Transactions on Education,* vol. 54, no. 3, pp. 356-365, 2011.

[12] C.-S. Lee, J.-H. Su, K.-E. Lin, J.-H. Chang and G.-H. Lin, "A project-based laboratory for learning embedded system design with industry support," *IEEE Transactions on Education,* vol. 53, no. 2, pp. 173-181, 2010.

[13] J. W. Bruce, J. C. Harden and R. B. Reese, "Cooperative and progressive design experience," *IEEE Transactions on Education,* vol. 47, no. 1, pp. 83-92, 2004.

[14] M. Lande and L. Leifer, "Prototyping to learn: Characterizing student prototyping activities," in *Proceedings of ICED 09, the 17th International Conference on Engineering Design*, Stanford University, California, United States, 2009.

[15] D. A. e. a. Kolb, Experiential learning: Experience as the source of learning and development, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[16] J. R. Dixon, "New goals for Engineering Education," pp. 55-62, 1991.

[17] R. S. Adams, J. Turns and C. J. Atman, "Educating effective engineering designers: The role of reflective practice," *Design Studies,* vol. 24, no. 3, pp. 275-294, 2003.

[18] P. Stuart, H. Dale and B. Sharyn , "The learning outcomes of an online reflective journal in engineering," in *Proceedings Ascilite*, 2008.

[19] K. S. Kearney, R. Damron and S. Sohoni, "Observing engineering student teams from the organization behavior perspective using linguistic analysis of student reflections and focus group interviews," *Advanced in Engineering Education,* vol. 4, no. 3, 20015.

[20] T. D. R., "A general inductive approach for analyzing qualitative evaluation data," *American Journal of Evaluation,* vol. 27, no. 2, pp. 237-246, 2006.

[21] S. A. Corbin. J, Basics of qualitative research: Techniques and procedures for developing grounded theory, Thousand Oaks, CA: Sage Publications Inc. , 2008.

Appendix A

Problem Statement for the Mini-Project

The purpose of this exercise is to design a control system that regulates the speed of a motor based using an Arduino and Rasberry Pi, along with an ultrasonic sensor and camera. The system should operate as follows:

- When started the motor does not run.
- When the camera detects that a green LED is lit, the motor runs with the speed determined by the ultrasonic sensor. If no object is detected, the motor runs at maximum speed. If an object is detected, the distance d in centimeters is calculated, and the motor runs so that the wheel rotates at d=10 radians per second. The desired speed should also be displayed on an LCD screen.
- When the camera detects that a red LED is lit, the motor stops, and does not start again until green is LED is detected.
- If no LED is detected, the system should operate in the mode corresponding to the most recently seen LED (i.e the wheel spins if the last LED seen was green, it is stopped if the last LED seen was red).

Appendix B

Prompts from the Weekly Reflection Logs

Prompts from the reflection logs filled out by students when working on the introductory exercises are as follows

1. Goal for the week.
2. What did you know about Arduino and Pi Computers[1]?
3. List all resources and what specifically you used or learnt from that resource to complete the challenge exercises.
4. Compile a list of all documentation created. Provide file name a short description of that file.
5. Provide an example of something that you would do differently or you could improve upon during the course of this exercise.
6. On a scale of 1-5, what is your comfort level with Arduinos and Pi computers after going through this exercise? (One being least comfortable and five being most comfortable).

Prompts from the reflection logs filled out by students after they start working teams

Weekly Overview

1. What goal(s) did your team set for this week?
2. What responsibilities were assigned to you for the week?
3. Rank your team in one of the following categories
    a. Pseudo Team: Performance below the level of an average member
    b. Potential Team: Struggling slightly above the level of the average member
    c. Real Team: Had complementary skills; is committed to common purpose, goals, approach, holds each other mutually accountable
    d. High Performing Team: Deeply Committed to one another's personal growth and success.
4. What evidence supports your ranking?

Reflection on your Learning

1. Did you have any "false starts" or did you begin down a path only to have to turn back? Describe in detail what happened. For example, what specific decision led you to the "false start"? If there were no false starts, why do you think you were able to progress so smoothly? Give a specific example.
2. In order to resolve problems that you encountered, what debugging experiments did you develop or use? Specifically, what experiments or setup helped you work your way backwards to discover the root of the problem that ultimately led to a solution?

Reflection on your Team and Team Work

1. Did your team come together this week to integrate parts of the project? If so, describe in detail what parts were integrated? Use the +/0/- scale to rate how successful integration of subsystems went?
2. Rate Success on a +/0/- scale
3. What about working on a team made it more difficult to resolve problems? What about working as a team made it easier to resolve problems?

[1] Replaced with OpenCV, Simulation, or Processor Communication when students working on those exercises

Appendix C

Design Case Study

The purpose of this project is to further advance efforts in manufacturing working antennas using 3D printing techniques. The anticipated result is a working GPS antenna manufactured using 3D printing techniques (leveraging the ADAPT center at Mines).

Project Description

In today's RF industry, rapid prototyping gives a company advantages over the competition in terms of quicker design maturity by providing the ability to perform multiple iterations of a design until optimized faster. 3D printing of antennas can improve our current process of rapid prototyping and increase our advantage over the competition. Colorado School of Mines is working with Ball Aerospace to advance our 3D printing capabilities through the ADAPT Center. This project will leverage from the 2016-2017 senior design team's (anTEAMa) final results and will focus on optimizing the current RF and mechanical design, surveying 3D printing vendors early on for cost and schedule, procurement of materials, building and testing antenna. The final result will be a working GPS antenna manufactured using 3D printing techniques.

The case above is an example of a senior design project. How would you approach this project? What steps would you take or processes would you use? What challenges would you anticipate? Focus on the describing your general approach, rather than explaining or designing a particular solution. Assume you have two semesters and reasonable resources available to complete the project.