6-2020

Work-in-Progress: Implementing Sophomore Cornerstone Courses in Electrical and Computer Engineering

Branimir Pejcinovic
Portland State University, pejcinb@pdx.edu

Melinda Holtzman
Portland State University, holtzman@pdx.edu

Phillip Wong
Portland State University

Citation Details

Pejcinovic, B., & Holtzman, M., & Wong, P. (2020, June), Work in Progress: Implementing Sophomore Cornerstone Courses in Electrical and Computer Engineering Paper presented at 2020 ASEE Virtual Annual Conference Content Access, Virtual On line. https://peer.asee.org/35648

ASEE'S VIRTUAL CONFERENCE
At Home with Engineering Education JUNE 22 - 26, 2020 #ASEEVC
Paper ID #29675

Work-in-progress: Implementing Sophomore Cornerstone Courses in Electrical and Computer Engineering

Prof. Branimir Pejcinovic, Portland State University

Branimir Pejcinovic received his Ph.D. degree from University of Massachusetts, Amherst. He is a Professor and former Associate Chair for Undergraduate Education at Portland State University, Electrical and Computer Engineering department. In this role he has led department-wide changes in curriculum with emphasis on project- and lab-based instruction and learning. His research interests are in the areas of engineering education, semiconductor device characterization, design and simulation, signal integrity and THz sensors. He is a member of IEEE and ASEE.

Dr. Melinda Holtzman, Portland State University

Melinda Holtzman received her Ph.D. in Electrical Engineering from the University of Nevada, Reno. She is a Senior Instructor in the ECE department at PSU.

Mr. Phillip Wong, Portland State University

Phillip Wong received an M.S. degree in electrical engineering from Carnegie Mellon University in 1990. Since then, he has been with Portland State University, Oregon, USA, where he is currently the ECE Lab Coordinator and an instructor.

Work-in-progress: Implementing Sophomore Cornerstone Courses in Electrical and Computer Engineering

Many engineering programs have significant project- and design-based courses in the freshman and senior years. Freshman courses usually serve a dual purpose: a) making engineering a more attractive study option for undecided students, and b) introducing the basics of technical and non-technical skills, such as teamwork and project planning. On the other hand, seniors doing their capstone projects are expected to perform at a more proficient level, one that would demonstrate their readiness for entering the workforce as engineers.

As will be discussed below, the project component in our freshman year-long sequence evolves from minimally structured projects, like designing and building Rube Goldberg contraptions, to more complex microcontroller-based projects where formal tools for teamwork and project management are introduced. However, we do not yet enforce strict adherence to procedures and processes. This is unlike the senior capstone, where the expectation is that students will not only be familiar with these methods but will use them effectively throughout their projects. This presents an obvious problem: what happens in the middle two years? Clearly, there is a very long gap, and unless there are more courses reinforcing the initial learning, it is likely that students will not develop further and will tend to forget even the basics of their teamwork and project management skills. Our answer to this problem has been the introduction of a two-quarter course sequence in the sophomore year: ECE 211 Introduction to Design Processes and ECE 212 Introduction to Project Development. For brevity, we will call this sequence the Cornerstone courses. We have provided an introduction to these courses in [1], but at that time we were still in the middle of our first implementation. In this work we will focus on the details of the course design, assessment used, and lessons learned from the first three offerings of the courses.

In the following sections we will discuss:

I. Overall ECE curriculum at our university (to provide proper context),
II. more detailed course learning outcomes and descriptions,
III. assessments used and results, and
IV. discussion of our observations so far.

I. ECE curriculum

For freshman-level students, the ECE 101, 102, and 103 sequence during the first year is their primer to engineering design and project management [2]. ECE 101 Exploring Electrical Engineering is a course that gives new students insight on what electrical and computer engineering is about and what opportunities are available for them if they choose to study in this field. The class is lecture-based, but faculty and practicing engineers from industry are often invited to give presentations on their work. In addition, students test simple electrical circuits during weekly labs. Finally, there is a major class project that involves teams of four-to-six students working on the design and construction of a “Rube Goldberg” machine that accomplishes a stated task while incorporating mechanical elements and some electrical components. At this point, the project management (PM) requirements are loosely structured and formal teaching of PM methodology is minimal.

The follow-on is ECE 102 Engineering Computation, which combines elements of a traditional engineering analysis and problem solving course with an accelerated introduction to using MATLAB as a calculation and programming tool [3]. Teams of students work on a well-defined, term-long hardware interfacing project. They perform background research, build a working prototype model, and write their own MATLAB program to control a LabJack measurement and automation unit that connects to the project hardware. An example of a prior project is building a model of a street intersection with working traffic lights and sensors. Whereas ECE 101 students managed their projects on an informal basis, in ECE 102 they are expected to employ basic project management techniques, such as a kanban board to track daily progress. The kanban board is implemented online using the Trello website, with weekly assessments by teaching staff to provide feedback. A final report and in-person demonstration of the completed project are required and assessed using a set of rubrics.

The first-year sequence culminates in ECE 103 Engineering Programming, which is a formal programming course that uses the C language, along with a very brief exposure to C++ and x86 assembly code. The organization of the course is similar to ECE 102, with lectures, labs, and a final project. Using the CATME Team-Maker service, teams are formed after students fill out a skills and schedule survey. Teams may choose from a set of available projects such as an LED multiplexed clock or a music synthesizer, among others. Unlike ECE 102, the projects here are more open-ended with fewer prescribed rules, so students can be more creative with their own design elements. More ambitious teams can elect to create their own project and use an alternate controller like an Arduino instead of a LabJack, if they wish. Students are still required to use Gantt charts and Trello for basic project management. However, teams now must also write a proposal to get approval from the instructor for their design and write several progress reports that precede the final report. Expectations for using Trello as a kanban board are also raised, with assessments being tightly focused on weekly progress and individual participation. At the midpoint and the end of the project, students are asked to perform a peer evaluation using CATME, which provides the instructor and the team members feedback on team dynamics and individual contributions.

The ECE 103 course offers a set of labs that contain a mix of general C programming exercises and hardware interfacing. Teaching staff are on hand during the lab to provide immediate feedback and guidance, especially when they introduce the ESP32 microcontroller to students. The ESP32 is a low-cost 32-bit system on a chip that has integrated Wi-Fi, Bluetooth, and I^2C/SPI capabilities. During the lab exercises, students learn how to use ESP32 and compatible

Arduino programming libraries to interface with sensors and display modules. While these labs are currently optional, they will be made mandatory in the next academic year. The labs allow students to see a practical application of their programming skills, which establishes a firm foundation for the second-year Cornerstone courses, both of which use an ESP32 board as the primary microcontroller for more advanced projects.

In their fourth and final year, all students participate in our Capstone three-quarter sequence. These courses have been very successful over the last two decades [4]. In short, student teams work on real-life projects that are proposed by local industry, and also have a faculty mentor. During the first quarter, students learn more about project management skills, teamwork, and documentation requirements. They also have a hands-on practicum project during which they have to design and implement some interesting product of their choosing. In this way, they become familiar with the prototyping resources available, and practice the technical and PM knowledge from their coursework. In the second and third quarters, students work on their own but in close contact with faculty and company mentors.

II. Cornerstone courses

We had two overarching goals for our Cornerstone courses:

1. Teach students design and project development well before senior Capstone projects.
2. Integrate various strands of electrical and computer engineering through experiential learning.

Given that there was little room in our freshman and junior year curriculum, the only option was to add a sophomore-level course. This means that we will have to rely on individual courses in the junior year to carry on the development of student skills and abilities. For the second goal, we decided to use advanced IoT-ready microcontrollers as a standard tool. This decision has affected our freshman courses because we will now require that some basic skills related to programming be more targeted towards microcontroller programming.

One issue that all engineering programs face is an overcrowded curriculum with constant tension between the need to cover ever-expanding technical topics and, at the same time, teach students the “soft skills” that they need to be effective engineers. This ties into accreditation, with ABET issuing a new set of criteria for evaluation of engineering programs [5]. One new item that ABET explicitly calls for is the development of teamwork and project management skills:

> *Criterion 5: “an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives”*

Most programs are constrained to 120 semester hours or, equivalently, 180 quarter hours for their overall program. This limited the number of credits that could be assigned to this sequence, and we decided to make ECE 211 a 1-credit course and ECE 212 a 2-credit course. We will be re-examining this decision at the end of this academic year.

We have formalized our course goals in a set of six learning outcomes:

1. Develop requirements and specifications based on stated need.
2. Find information for the design of a product that meets customer needs.
3. Perform a functional decomposition for a given design.
4. Communicate effectively: create a design log, deliver progress reports (oral or written). document product design specification, and write a final report.
5. Implement basic project management techniques.
6. Complete and demonstrate a project that satisfies a specific need.

In addition to the specific learning outcomes given above, these courses were designed to advance student learning as they move from freshman to sophomore level courses. The main areas we considered for this advancement are related to design, project management (PM), and teamwork, as described below:

1. Developing projects:
   Freshman: Show basic skills in breaking down assignments into team tasks
   Sophomore: Develop projects from a starting idea and functionally decompose them
2. Applying Scrum PM:
   Freshman: Show evidence of planning for their projects
   Sophomore: Apply Scrum PM more fully
3. Using project planning tools:
   Freshman: Learn the basics of project planning tools
   Sophomore: Use project planning tools effectively
4. Effective teamwork:
   Freshman: Run projects with minimal team conflict
   Sophomore: Demonstrate a deeper understanding of team dynamics

One of our implicit goals for Cornerstone courses is to give students freedom of expression, so we impose few restrictions on the type of products (projects) to be developed. Similarly, students are encouraged to “think big” and not to be afraid of failure. Given the course infrastructure that we discuss next, the course instructor has good insight into how much effort, thought, and planning goes into individual projects. He/she can determine relatively easily if teams are failing due to lack of effort or have, for example, run out of time to complete an ambitious project. We also encourage teams to enter competitions such as Cleantech Challenge [6], where teams pitch their project ideas and the most successful ones get additional funding.

A unique aspect of our Cornerstone courses is the use of the Scrum PM approach. Previously, we have described how we use and assess it across the ECE curriculum [7], [8]. We largely follow the usual Scrum organization [9] but have made some adjustments to accommodate the educational environment, including [8]:

1. Roles of Product Owner and Scrum Master are flexible and adjusted to specific level and course contexts.
2. Students are required to use a kanban board to track and document their planning and progress in implementation.
3. Teams are formed by instructors.
4. Daily Stand-up meetings are held 3 or 4 times per week at regularly scheduled times.
5. Gantt charts are used for overall project planning and to visualize Sprints.

We have found that upper-division students make for excellent Scrum Masters (team leaders) and each Scrum Master is assigned two or three teams to work with.

The rest of this paper is based on implementation and analysis of two offerings of the Cornerstone courses: one in Fall 2018 and Winter 2019, and the other in summer 2019. The first one had 58 students in 15 teams, while the second one had 19 students and 5 teams. At the time of writing, the third offering is underway. Most of the results presented below are from the first offering. Our summer session lasts only 8 weeks, which means that activities have to be compressed and the schedule is less than ideal. Nonetheless, we managed to maintain all of the essential components of the courses in the summer term, which is an offering particularly beneficial to transfer students who transfer from local community colleges as juniors and have not taken equivalent courses.

Scheduling

From our stated goals and aspirations, it is clear that there is a lot going on in these quarter-long classes. Therefore, careful planning of activities is required. Table I illustrates what our current schedule looks like for the first course:

Table I: Weekly plan for ECE 211

Week	Topic / Activity	Deliverables
1	Introduction, project selection	In-class project brainstorming
2	Teams and teamwork	Lab 1 demo (ESP32 setup); team contract
3	Project management (Scrum)	Lab 2 demo (interfacing)
4	PM tools (Trello)	Lab 3 demo (sensors and displays)

5	Functional decomposition	FD sketches; sprint planning
6-7	Practicum	Demo of working “product”
8-9	Design	Practicum sprint review and retrospective
10-11	Project planning	ECE 212 project proposal report

It is important that students are given opportunities to practice the things we discuss in class related to teamwork, project planning and management, design, and the technical skills of using microcontrollers. As the schedule in Table I shows, we have inserted a two-week long practicum for this purpose. Students are given a list of potential projects which have good online documentation and need only minor modifications to make them work. For example, these two projects have been very popular [10], [11]: one is to build a 3D printed mailbox that notifies you when you received an email, and the other is a weather station box that uses internet resources to report on current and future weather. These are straightforward projects, but teams usually have to adjust the available code so that it runs on the ESP32-based microcontroller [12].

Similarly, most students are not familiar with solid-modeling software so they have to learn it. Usually, other technical problems arise as teams start working on the projects. Finally, teams get to practice division of labor and planning of activities. The main criterion is that whatever they build must demonstrate the basic functionality which was given in the project description.

Note that the labs listed in Table I are the same as the ones we have described above in the section dealing with ECE 103. It may seem redundant to repeat these, but a large percentage of our students are transfer students who typically do not have the relevant experience. Students who have taken these labs before are asked to tutor those who did not, and the whole team has to demo the lab to one of the class Scrum Masters. Currently, labs 1-3 are optional in ECE 103, but will be required in the next academic year. At that time, we will design several new labs involving IoT and advanced microcontroller features. This will give students who took the labs previously in ECE 103 the chance to expand their skills.

While this is a 1-credit course, we have been scheduling 3 hours of in-class time, typically on Monday-Wednesday-Friday. This way students have times blocked out on their schedule so that they can hold their team meetings during some part of these 3 hours. Most teams require additional meeting times, and CATME is very useful in setting up teams with compatible schedules. Since Cornerstone courses are project based, there is not much time spent in traditional lecture mode; instead, we have designed activities where teams put everything into immediate practice.

Table II. Weekly plan for ECE 212

Week	Activity	Deliverables
1	Revised team contract; detailed project planning; 1st sprint planning	Gantt chart; level-0 (L0) and level-1 (L1) functional decomposition; full product backlog; 1st sprint backlog
2-5	1st and 2nd sprint	Sprint reports (review, retrospective, plan); early prototype demo; CATME evaluation
6-9	3rd and 4th sprint	Sprint reports (review, retrospective, planning)
10-11	Final prototype demos	Working prototype and poster; project report

The first week of ECE 212 is devoted to project planning, culminating in a Sprint Planning session when all the details of the first sprint are decided and entered into Trello. A more detailed overall plan that encompasses the whole quarter is also developed and is represented in a Gantt chart. Since we are using Scrum as our PM methodology, teams break up their work into 2-week long sprints. At the end of each sprint, teams hold a Sprint Review, Retrospective, and Planning session. Briefly, the Sprint Review looks back on the just-finished sprint and asks if all tasks have been accomplished, and if anything will be carried over into the next sprint. It deals with technical aspects of the project. The Sprint Retrospective is devoted to how well the team functioned and how to improve the teamwork aspect. The main concern is about the process. Finally, Sprint Planning sets all the goals, tasks and deadlines for the next upcoming sprint.

Each team is assigned an upper-division student who acts as Scrum Master. As in real-world Scrum teams, Scrum Masters lead all the meetings, including so-called daily standups. They help with identifying resources for team projects, but are not involved in day-to-day technical work. Scrum Masters also provide periodic feedback to teams through evaluation of the teams' Trello boards and by filling out a bi-weekly PM evaluation rubric, as discussed below. The course instructor visits daily standups and Sprint Review, Retrospective, and Planning sessions and provides his/her own feedback to all teams.

At the end of the term, teams present their product prototypes along with a poster. This demo is patterned after our Capstone demo day so that students get a preview of what is expected in their senior year. Projects are finished by writing a formal report.

In summer term, ECE 211 was compressed into one week, meeting three hours a day for five days. The practicum was reduced to a “basic skills demo”, where teams had the option to either 3D-print something, or complete one of a set of microcontroller exercises (somewhat more advanced than the ECE 211 labs). These options were chosen to develop basic skills typically needed in the ECE 212 projects. ECE 212 was seven weeks, with the first week devoted to the basic skills demo and project planning, followed by three 2-week sprints to complete the project. Some of the documentation requirements were reduced, such as not requiring a poster, but all the essential elements were retained.

III. Assessment

Assessment of learning in project-based courses is challenging due to their holistic nature and complex structure [13]. Assessment is essential for course improvement, but it also needs to be structured well so that it does not send misleading signals to students with respect to the importance of various components. Given the structure of the Cornerstone courses, we also have to be flexible, especially if we want to encourage students to experiment and not be afraid of failure. We would also like to have different ways of assessing and evaluating learning outcomes, which would enable triangulation. With all this in mind, we have decided to evaluate these components:

a) Teamwork
b) Project management
c) Design and experimentation
d) Written and oral communication

These outcomes are evaluated using artifacts, e.g., written reports, or by direct observation of teamwork by the Scrum Masters and instructor. In the sections below, we discuss the assessment tools that we use.

Teamwork: team contracts

Experience has shown that the reason most teams fail is not technical incompetence, but internal team conflict. There are many possible causes for this failure, and we try to preempt some of the most common problems by having students develop and sign so-called team contracts. Students are taught some basics of team dynamics and how to avoid or manage potential conflicts. Well-known phases of team development [14]: forming, storming, norming, performing and adjourning, are discussed in ECE 211. Our team contract template was based on [15] but simplified and made more applicable to the Scrum-like team environment. The main components are:

- Team procedures: time, date and place for Scrum events; decision-making policy; setting up agendas; record keeping; writing reports

- Team expectations: work quality, team participation, personal accountability (attendance, responsibilities, communicating with the team)
- Consequences for failing to follow procedures and fulfill expectations

Each team is asked to develop an initial team contract in ECE 211 before they embark on the practicum project, and the instructor gives feedback on this version. Teams submit a revised version at the beginning of ECE 212, at which time they have worked together for well over a month and had to deliver several artifacts as a team.

Team contracts are judged on completeness, specificity, and general quality of entries. So far, teams are taking this task seriously and very few contracts are poorly done. However, we have not been able to establish how helpful these contracts are in preventing undesirable team behavior. For example, all teams will specify the consequences for failing to live up to the contract terms, but students are still loathe to implement these consequences in practice. Fortunately, we do have other ways of identifying potential problems, such as observation of team dynamics by Scrum Masters, who can attempt to resolve problems with the team or bring in the instructor in the most difficult cases.

Teamwork: peer evaluation

In our department we use CATME [16] in many courses to form teams and to assess teamwork based on peer evaluations. We have found it to be a very useful and reliable tool for courses across the curriculum [7], [8] . In each of our Cornerstone courses we use peer evaluation twice. In both courses, the first evaluation is primarily diagnostic, as we are trying to find out if there are any teams with problems that would require intervention by the instructor or Scrum Master. In that sense, this one is formative in nature. The second one, however, is summative because we use its results to determine an individual's contribution to the team and the corresponding grade.

Most students take this evaluation seriously, but we still find that some students are hesitant to provide an unvarnished evaluation of their peers. This is why we try to triangulate these results with other tools, especially feedback from Scrum Masters and instructor's observations. So far, we have found almost perfect correlation in terms of identifying troubled teams. One area that will need improvement is training students in the use of CATME. Students who have taken our freshman sequence ECE 101/2/3 would have received some of this training, but our transfer students have not. Our plan is to utilize CATME's online resources and provide some of the most effective training techniques explained in [17].

Project management

Project management is surprisingly difficult to assess. There does not seem to be any developed and published rubrics, or other means to directly assess and evaluate it in the context of engineering coursework. What can be done relatively easily is to ask students to report on how they set up and performed their project management. This has the obvious problem of the reliability of self-reports, where students do not have appropriate training to evaluate themselves and may overstate performance. Another way could be to ask for some artifacts, such as the Gantt chart, to be developed at the beginning and revised at the end with the hope that we can assess what happened in between these two discrete points in time. However, we have no insight into the actual implementation of this plan.

Given that one of the advantages of using Scrum for PM is the transparency that it provides [7], [8], and that our Scrum PM works on 2-week intervals (sprints), this presents opportunities for more frequent assessment and for providing meaningful feedback.

We use the following ABET definition as a starting point in the assessment of project management:

> *"An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives." [5]*

Table III summarizes the performance criteria for our newly-developed PM rubric[1].

Table III. Project management rubric

Criterion	Performance Indicators	Artifacts / Evaluation
A. Project planning	What is the quality of planning documentation, timeline, decomposition of the product, and stated requirements?	Reports on each of the four Sprint Review, Retrospective, and Planning sessions
B. Project implementation	How well is the team executing its plans, individual tasks, and is it meeting the deadlines?	Direct observation
C. Team functioning	Are the team structure and roles well defined? Does the team communicate well? Is there a sense of team spirit?	Direct observation

The performance level scale used is standardized to: Exemplary (4), Proficient (3), Developing (2), Beginning (1). For each performance criteria, two or more performance indicators have been developed.

[1] More detailed rubrics including performance metrics for each criterion are available from the authors.

Scrum Masters fill in this rubric for each sprint and share their assessment with their teams. The instructor does one assessment at the end of the term. Given that teams track their planning and implementation activities using Trello [18], [8], Scrum Masters also give feedback on how well teams are maintaining their Trello boards.

Data for the instructor's evaluation of project management using this rubric is shown in Figure 1 for the ECE 212 course in Winter 2019. Note that we used intermediate scoring so that "2.5" corresponds to a level between "Developing" and "Proficient".

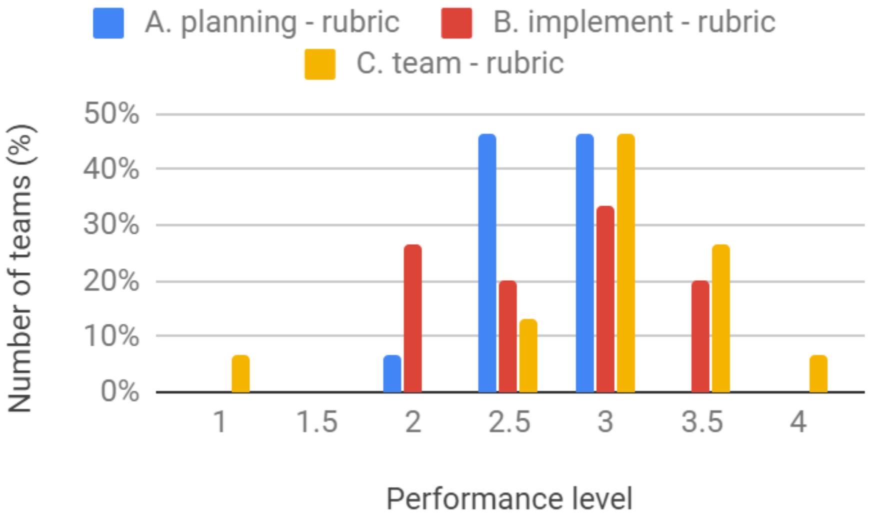

Figure 1. Percentage of teams in ECE 212 that have attained a given performance level using our project management rubric. Levels are: Beginning (1), Developing (2), Proficient (3), and Exemplary (4). Intermediate levels are allowed.

Ideally, we would hope to see all teams achieve Proficient or Exemplary level. Realistically, however, we are looking to minimize the number of teams at either Beginning or Developing level. We can make a few qualitative observations regarding the data in Figure 1:

1. Implementation (criterion B.) had the largest number of teams at the Developing level and fewest at the Proficient level, indicating a significant potential for improvement. This criterion currently includes:
    a. Team meets regularly
    b. Plans are updated regularly
    c. Tasks are specific, detailed, have responsibilities assigned with deadlines, and define when a task is done (who, when, what)
    d. Activities follow a plan
    e. Tasks are updated regularly and progress checked
    f. Team meets deadlines

Observed problems may be related to poor planning (criterion A.), but we suspect that some additional external factors, such as exams and deadlines, intervene and detract from planned project tasks.

2. Planning their project (criterion A.) is also challenging for teams. However, these are sophomore students or transfer juniors, and we would expect their performance to be somewhat lower than proficient.
3. Team functioning (criterion C.) was not a major problem. Of the 15 teams, only one had a serious problem with team dynamics that required the instructor's intervention.

Based on this evaluation, we will place more emphasis on teams' planning of activities and have Scrum Masters pay closer attention and give more frequent feedback regarding teams' implementation of their plans.

These rubrics, as well as ways of using them, are still a work in progress. Keeping teams on track with their project management is still a challenge. Some students view it as an unnecessary burden, which can detract from their motivation. Teams do show improvement over the course of four sprints, but we have to keep reminding them of the importance of this segment of their project. Our hope is that this experience will be reflected in more effective teamwork and project management in their capstone projects, but we do not have evidence of this yet.

Design component

Given the time constraints, we can only introduce the very basics of design process in ECE 211, and then follow that up with prototype production in ECE 212. This necessitates that we evaluate this component mostly in a formative fashion, i.e., as a stepping stone towards a fuller understanding of the design in our capstone courses. We used the ABET definition of design to develop our assessment rubric:

> *"An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors."*

Note that unlike the freshman courses, here we stress that each project must meet a need, which is determined and described by each team.

Performance criteria we selected are:

A. Identifies and follows a logical and orderly design procedure.
B. Systematically develops, compares and ranks design alternatives to arrive at a final solution.
C. Creates a final solution that satisfies all requirements and constraints identified in formulating the design problem.

D. Justifies design decisions using analyses based on appropriate engineering and/or scientific principles.
E. Considers, where appropriate, factors: public health; safety and welfare; global, cultural, social, environmental, and economic.
F. Supports the design process with appropriate engineering documentation and references.

For each performance criteria, one or more performance indicators have been developed and our standard performance level scale was used: Exemplary (4), Proficient (3), Developing (2), Beginning (1)

This rubric was used to evaluate final team reports. Overall, teams performed at levels close to Proficient, which we consider satisfactory for sophomore students. Areas in need of improvement include:

a) Team should report on the details of their design process and justify some of their decisions and requirements.
b) Teams should provide better description of their L0 and L1 decomposition (see [19] for explanation of L0 and L1 decomposition).

Future improvements will include:

1. State a more detailed set of requirements for the final report
2. Have Scrum Masters encourage teams to start writing the outline of their final report earlier, and give feedback on those outlines

We have made one additional observation regarding testing. Testing is not always considered during the design stages, but we require students to specify at least two functions for their product as well as performance metrics that they will use to determine if their product meets the functional description. This concept is hard for students to grasp initially and is further compounded by the common confusion regarding the difference between debugging and testing. Students use these terms interchangeably and, in general, prefer to discuss debugging instead of testing. In our context, debugging is the process of identifying and removing errors from hardware or software. Testing, on the other hand, is "to take measures to check the quality, performance, or reliability of (something), especially before putting it into widespread use or practice" [20]. We are working on examples that we hope will illustrate this difference and also reinforce it during the prototype development stage.

Demos

Teams are required to do an in-class demo of a working prototype after the practicum project in ECE 211 and at the end of their final project in ECE 212. We have developed a brief rubric to guide evaluation of these events, which contains the following criteria:

A. Overall quality of the demo presentation

B. Build quality of the prototype
C. Project discussion
D. Extra credit for special, unexpected, or original product features

The first set of demos proved very informative because many teams experienced the same problem: they waited too long to integrate components into the final product. Teams discovered that even though all components seemed to work fine individually, this is no guarantee that the system will work as expected. This led to some disappointment during demo time but was also good experience for later, bigger projects.

Similar observations were made in the summer session, where students did not have the advantage of a practicum experience, but were warned about the need for integration and troubleshooting the complete product. Four of the five teams had successful demonstrations, but one team had some problems stemming from this issue.

Organization and delivery of the demo is also an opportunity to assess students' communication skills. Overall, we find that most teams perform well in this category.

Communication

Written and oral communication is assessed by using two rubrics. For the former we use written reports and for the latter we use team presentations of proposed projects. Scoring rubrics used for grading these include:

A. Written report rubric criteria: Content, Organization, Vocabulary and Grammar, and Mechanics (formatting, spelling, proofreading)
B. Presentation rubric criteria: Organization, Visual Aids, Technical Content, Posture and Tone, Handling of Questions, and Effective Use of Time

We have also done a more detailed analysis using a rubric that is used for assessing program learning outcomes, but that rubric will be reported elsewhere. Based on that evaluation, students came close to the "Proficient" level in their writing communication skills. We consider this a good result given that this is a sophomore class. Use of this extended rubric uncovered one glaring problem regarding students reporting of test results. Teams preferred to write extensive descriptions of their test procedures and results but did not think to present them in a nice, technical fashion using, e.g., charts or tables. In the future we will be making this a specific requirement.

We have observed that teams perform reasonably well in their oral presentations. We impose a strict limit of 5 minutes total and 15 seconds per slide. Because slides advance automatically, teams are forced to plan ahead and practice. Areas of improvement include utilizing more visually appealing content, reducing the amount of text, and using more visible fonts.

IV. Discussion and conclusions

We previously reported some of our very early observations in [1]. Based on the discussion presented above, we make the following observations:

1. Most teams do well in successfully completing and demonstrating their projects
2. CATME can reliably identify teams with problems
3. The usefulness of team contracts is to be determined
4. Teams can deliver decent presentations, but
5. Written reports are more challenging
6. We need to teach students to use charts and tables instead of text to convey technical information
7. Teams improve over the course of two quarters in the use of PM tools, but need constant monitoring (provided by Scrum Masters) and feedback
8. Students are enthusiastic about the hands-on project-based class and gain satisfaction from designing and building their own projects

In summary, we have taught these Cornerstone courses twice with good results and are currently in the middle of the third offering. We have described the details of the course implementation, including scheduling and how to implement Scrum project management.

Students have shown the creativity, motivation, and growth that we had hoped for, and student response to the courses has been positive. To assess our goals more rigorously, we have developed a set of rubrics which we have described in detail. Results of the initial assessment are positive and have provided insights that we will use to improve the Cornerstone courses. For example, we will insist on more formal planning of testing, and better presentation of prototype testing results. We will also emphasize the difference between debugging and testing, which are often confused by students. In the future we also plan to expand our discussion of the design process and to introduce topics on ethics in engineering.

The novelty of our approach is in:

1. Introducing a project-based design course sequence in the sophomore year which emphasizes project management with some rigor
2. Using the Scrum methodology for project management
3. Using upper-division students as Scrum Masters
4. Providing frequent feedback during all phases of the project

In conclusion, we believe that our approach is effective in accomplishing our curricular and learning goals, and can be replicated elsewhere.

References

[1] B. Pejcinovic, "Introducing Cornerstone Courses into Electrical and Computer Engineering Curriculum," 42nd International Convention MIPRO, 2019, pp. 1427–1432, DOI: 10.23919/MIPRO.2019.8756847.

[2] P. Wong, M. Holtzman, B. Pejcinovic, and M. Chrzanowska-Jeske, "Redesign of Freshman Electrical Engineering Courses for Improved Motivation and Early Introduction of Design," in ASEE Annual Conference and Exhibition, Vancouver, Canada, 2011.

[3] B. Pejcinovic and P. Wong, "Evolution of an Introductory Electrical Engineering and Programming Course," in 2017 ASEE Annual Conference & Exposition, Columbus, OH, 2017.

[4] M. Faust, A. Greenberg, and B. Pejcinovic, "Redesign of senior capstone program in electrical and computer engineering and its assessment," in 2012 Frontiers in Education Conference Proceedings, 2012, DOI: 10.1109/FIE.2012.6462390.

[5] ABET Criteria for Accrediting Engineering Programs, [Online]. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/, [Accessed Feb. 3, 2020].

[6] Cleantech Challenge program at Portland State University, https://www.pdx.edu/clean-challenge/

[7] B. Pejcinovic, R. B. Bass, and P. Wong, "Assessing Scrum Project Management and Teamwork in Electrical and Computer Engineering Courses," in 2018 ASEE Annual Conference & Exposition, Salt Lake City, Utah, 2018. https://peer.asee.org/29822

[8] B. Pejcinovic, P. Wong, and R. Bass, "Adapting Scrum Project Management to ECE Courses," in ASEE Annual Conference & Exposition, 2019, Tampa, FL. https://peer.asee.org/32395

[9] The Scrum Guide, [Online]. Available: http://www.scrumguides.org/scrum-guide.html. [Accessed Feb. 3, 2020].

[10] Project 1: Gmail mailbox, [Online]. Available: https://learn.adafruit.com/gmailbox?view=all, [Accessed Feb. 3, 2020].

[11] Project 2: WiFi Weather Display, [Online]. Available: https://www.instructables.com/id/WiFi-Weather-Display-With-ESP8266/, [Accessed Feb. 3, 2020].

[12] Adafruit HUZZAH32 ESP32 Feather Board, [Online]. Available: https://www.adafruit.com/product/3405, [Accessed Feb. 3, 2020].

[13] D. Hellström, F. Nilsson, and A. Olsson, "Group assessment challenges in project-based learning – Perceptions from students in higher engineering courses," presented at the 2:a Utvecklingskonferensen för Sveriges ingenjörsutbildningar, Lund, Sweden, 2009, p. 6.

[14] B.W. Tuckman, "Developmental sequence in small groups," Psychological Bulletin. Vol. 63, no. 6, pp. 384–399, 1965. DOI:10.1037/h0022100

[15] Team contract template, [Online]. Available: https://cns.utexas.edu/images/CNS/TIDES/teaching-portal/Team_Contract.doc, [Accessed Feb. 3, 2020].

[16] M.W. Ohland, et. al., "The comprehensive assessment of team member effectiveness: Development of a behaviorally anchored rating scale for self and peer evaluation," Academy of Management Learning & Education, Vol. 11, no. 4, pp. 609-630, 2012.

[17] D. M. Ferguson, et al., "Evaluating the effect of different teamwork training interventions on the quality of peer evaluations," in 2018 IEEE Frontiers in Education Conference (FIE), 2018, DOI: 10.1109/FIE.2018.8658782.t

[18] Trello, [Online]. Available: trello.com, [Accessed Feb. 3, 2020].

[19] R. Ford and C. Coulston, Design for Electrical and Computer Engineers, Boston: McGraw-Hill Education, 2007.

[20] Definition of "testing", [Online]. Available: https://www.lexico.com/en/definition/test, [Accessed Feb. 3, 2020].