A Learning Science Foundation for Project-Based Learning in Engineering

Robert Niewoehner

US Naval Academy

Ed Crawley

Massachusetts Institute of Technology

Jean Koster

University of Colorado

Tim Simpson

Pennsylvania State University

ABSTRACT

Seeking to connect the cognitive sciences and teaching faculty, Susan Ambrose and her co-authors recently published, *How Learning Works: 7 Research Based Principles for Smart Teaching.*[1] Ambrose and her co-authors observed that cognitive and educational psychology was making great strides advancing the science of learning, but little of this science was impacting college classrooms. Ambrose *et al.* sought to connect effective teaching practices and cognitive psychology's advances in our understanding of learning and bring that science of learning into others' classrooms. Their book distills seven principles from the learning sciences, and then instantiates those principles with concrete teaching practices.

We find in Ambrose's work a substantiation of project-based learning in engineering, providing a foundation for understanding why this pedagogy works. Specifically, problem-based learning works because it naturally embodies all seven research-based principles of effective teaching and learning outlined by Ambrose and her co-authors. Appropriately executed, project-based learning implicitly complies with our students' ability to learn. We elaborate on four of Ambrose's seven findings and describe how the documented practices of emerging from the CDIO initiative instantiate those principles. Furthermore, Ambrose's principles suggest criteria by which we might justifiably identify *best practices* in project-based learning. This assessment may help promote and facilitate adoption of fine-tuned educational strategies within the CDIO framework. Furthermore, this will shift the arguments for project-based learning from appeals to intuition and trial-and-error to a more rigorous foundation built from the teaching and learning sciences.

KEYWORDS

Learning science, project-based learning, CDIO, cognitive psychology.

INTRODUCTION

Proponents of CDIO and project-based learning commonly hear the question, "What evidence do you have that it works?" In other words, photos of smiling student teams make nice presentations, but are these students really better engineers as a consequence? Formally, what body of research indicates that project-based learning represents a more effective pedagogy? Admittedly, while our disciplinary scholarship is data-driven, our pedagogy has depended largely on intuition and trial-and-error approaches based on what has (and has not) worked well in the classroom.

Seeking to connect the cognitive sciences and teaching faculty, Susan Ambrose and her co-authors recently published, *How Learning Works: 7 Research Based Principles for Smart Teaching.*[1] Ambrose and her co-authors observed that cognitive and educational psychology was making great strides advancing the science of learning, but little of this science was impacting college classrooms. College professors in other fields are unlikely to read cognitive psychology journals, or attend educational psychology conferences. Hence, this flourishing science of learning has been cloistered in an ivory tower on the other side of campus. Ambrose *et al.* sought to connect effective teaching practices and cognitive psychology's advances in our understanding of learning and bring that science of learning into others' classrooms.

We find in Ambrose's work a substantiation of project-based learning in engineering, providing a foundation for understanding why this pedagogy works. Specifically, project-based learning naturally embodies all seven research-based principles of effective teaching and learning outlined by Ambrose and her co-authors. Appropriately executed, it implicitly complies with our students' ability to learn. Our paper elaborates on four of Ambrose's seven findings and describes how the documented practices of project-based learning in engineering instantiate those principles. Furthermore, Ambrose's principles suggest criteria by which we might justifiably identify *best practices* in and project-based learning. This assessment may help promote and facilitate adoption of fine-tuned educational strategies within the CDIO framework. Furthermore, this will shift the arguments for project-based learning from appeals to intuition and trial-and-error to a more rigorous foundation built from the teaching and learning sciences.

Ambrose *et al* distill seven principles from the burgeoning scholarship of learning, which they pose as foundational insights for teachers/professors seeking to improve their students' learning. Their presumed audience is undergraduate faculty, as evidenced by their selected examples, but the principles are no less applicable to other learning contexts. These principles are:

1. Prior Knowledge: Students prior knowledge can help or hinder learning.
2. Knowledge Organization: How students organize knowledge affects how they learn and apply what they know.
3. Motivation: Students' motivation generates, directs, and sustains what they do to learn.
4. Mastery: To develop mastery, students must acquire component skills, practice integrating them, and know when to apply what they've learned.
5. Practice: Goal directed practice, coupled with targeted feedback, are critical to learning.
6. Student Development: Students' current level of development interacts with the social, emotional and intellectual climate of the course to impact learning.
7. Metacognition: To become self-directed learners, students' must learn to assess the demands of the task, evaluate their own knowledge and skills, plan their approach, monitor their progress, and adjust their strategies as needed.

A chapter of Ambrose's book is devoted to each of these principles, following a common structure. That structure flows from stories illustrating each principle, a diagnosis of the stories, a summary recounting of the research findings, and then research-informed suggestions for improving student learning.

While we believe that all seven of Ambrose's principles pertain to project-based learning, our paper will concentrate on four. In each case, we'll elaborate shortly on the principle, discuss how project-based learning complies, and finally how thoughtful application of the principle might improve those practices.

The CDIO construct is developed more fully in other sources [2]. We'd like to briefly clarify several important terms, where questions frequently arise. While project-based learning is an important element of the CDIO construct, appearing explicitly as CDIO Standards 5 & 6, it is but an element. (See reference [3] for the general goals and strategies of project based learning.) CDIO program standard #1 speaks of embracing engineering product development as the *context* for engineering education. *Contextual learning* we distinguish from project-based learning to emphasize that projects are designed not only with the goal of developing particular disciplinary knowledge but also the professional skills of engineers, such as writing, speaking, ethics, systems thinking. Where used, projects are set in a context which replicates the activities of an engineering enterprise doing the work of conceiving, designing, implementing, and operating engineering systems. Contextual learning is frequently embodied in hands-on, design-implement projects (project-based learning), but may also be found in learning activities that replicate business activities that do not require fabrication or fit the mold of "projects". Therefore contextual learning and project-based learning commonly overlap or coincide in CDIO programs, but project-based learning need not be contextual, nor contextual learning necessarily project-based.

Figure 1 depicts the foundational goal of CDIO, drawn from [2]. Specifically, we do not simply seek to move to the right in merely improving our students' learning of our disciplinary content. Our ambition is to move our programs up and right, simultaneously targeting deep learning of our disciplinary content *while* growing in the professional skills of engineers. This is how we believe problem-based learning varies in substance with a more narrow understanding of project-based, or project-centered learning, which may or may not target both disciplinary and professional development.

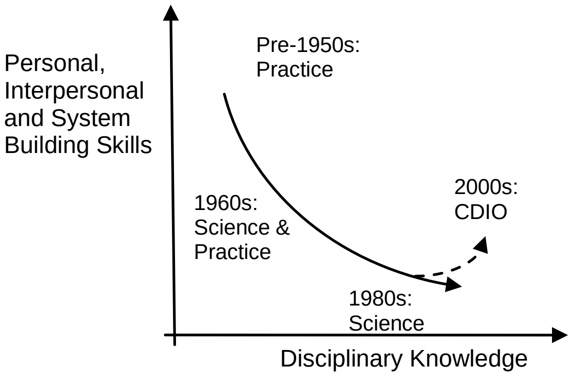

Figure 1. The Trajectory of CDIO's Desired Engineering Education Reform

In the sections that follow, we've picked four of Ambrose's seven principles to elaborate, exemplify, and then draw suggestions for excellent research-based practices in project-based learning, though many of our observations will also apply to contextual learning. We'll briefly address the three remaining principles at the conclusion.

OUR OBSERVATIONS

Principle 1- Prior Knowledge: Students prior knowledge can help or hinder learning.

Professors commonly assume that students have learned and retained basic skills from prerequisite courses, an assumption that may be further strengthened by the student's self-report. The issue is whether the Professor and the students have the same references; there may be a mismatch between knowledge that the students have or believe they have, and the knowledge the instructor expects for their course. Hence Ambrose's first principle:

Students prior knowledge can help or hinder learning

Activate prior knowledge

In order that their prior learning help in the new context, the professor must connect new content to prior knowledge and experiences. However, that prior knowledge must be appropriate to fit with new knowledge. Misconceptions and inaccuracies in prior knowledge may interfere with the learning of new material. Not all prior knowledge provides a solid foundation for new knowledge. Nevertheless students interpret new information based on their prior knowledge, whether accurate or false.

Students may not make connections to relevant prior knowledge, which makes new learning difficult and when prior knowledge is inaccurate or inappropriate, it will distort new learning.
Students must connect new knowledge to previous one in order to learn [4] Students commonly lack training in activating prior knowledge. Often minor prompts can activate relevant prior knowledge. This can often be done if the problem is stated in the context of applications or a design problem. Team-working with discussion sessions in capstone design play an important function in activating prior knowledge as students will often ask the question, "why?" Researchers call this process *elaborative interrogation*. Project based learning, especially in small teams, provide a fruitful environment where students can readily connect what they are learning to what they already know. Cross-department teams, where a team includes majors from more than one program, are advantaged here. A mechanical engineer might have studied electrical motors in his required EE courses, but carried misconceptions into her design team's work. A EE student with more depth in the subject might quickly recognize their teammate's confusion and correct it, providing insight which even a faculty member from another discipline might not appreciate. Mazur comments on the effective proximity of students to their peers' misconceptions [5].

Accurate but Insufficient Knowledge

In the case where the prior knowledge is insufficient but accurate, the new learning may become challenging. Students may have simply forgotten the details from prerequisite courses because the context of why the material is taught was not clear to the student. Engineers develop primarily declarative knowledge, or knowledge of facts and concepts that can be stated or declared.

A second knowledge is the procedural knowledge where knowledge is about when to apply a certain procedure, method or theory. With this knowledge students develop the skills when and how to apply their declarative knowledge.[6] In the self-directed learning environment of the design courses, students learn to recognize insufficient knowledge and can mitigate the deficiency. Further, the knowledge activation environment of a team can activate declarative knowledge in one team member and procedural knowledge in another student. A synergy of knowledge development is activated in project passed team learning. In a traditional lecture class it will remain a challenge to the teacher to mitigate insufficient knowledge negative reinforcement which is perceived punishment with bad grades [1].

Inappropriate Knowledge

Students may draw upon superficial information from other experiences and include a bias in their interpretation of a new knowledge. They may have been used to applying analogies for understanding a concept. Analogies have limits and it is essential to recognize these limits. Such a task requires careful analysis and students may not yet be at an intellectual point where they have clear references for such a comparison.

Knowledge is also related to group cultures or discipline cultures. Engineering students commonly learn English writing principally from English Departments that cherish particular writing styles. That writing style constitutes inappropriate knowledge for many aspects of engineering reports, particularly since argumentation is commonly deductively organized. In this case it is necessary to teach students the concise, precise and clear language requirements that engineers interpret in the desired context, and the inductive flow of technical reasoning. A good technique for testing engineering clarity of a text is to give team members and non-team members the draft to read and check whether they understood fully after first reading. Such an environment is provided by the design team organization in project-based learning.

Inaccurate Knowledge

Inaccurate knowledge is often disseminated through web based media and bragging rights of individual citizens. Such knowledge is not solid and the owners may change opinions on the fly. In the declarative knowledge of engineering inaccurate knowledge can often be corrected. Scientific claims in some domains, however may remain contentious. Such opinions may be misconceptions and are difficult to refute for many reasons. Reasons may be within limited scientific evidence or coming from groups with a specific agenda for their own benefit. In traditional engineering lectures the teacher may not provide sufficient proof that some concept residing in the students head is inaccurate or wrong.

In contrast, in engineering design the work focuses around technologies that are already proven to a set technical readiness level. Components that are designed to provide a defined function provide it or do not provide it. Students must verify properties and functioning, and validate the outcome. The student might have had a different understanding of how a system functions, but the results of her work will instill the correct knowledge in the student. This process is also often referred to as “experience”. It is common to see students presenting a technical design to an experienced machinist who tells the student at first sight “that does not work!” The next time the student returns, her system will probably “work”. Project-based learning is an excellent method to correct inaccurate prior knowledge.

Strategies to Activate Prior Knowledge

Project-based learning puts the learning requirements in the context of engineering technology. If the project is related to the specific field of engineering studies, the students with a pre conceived preference for that topic will be highly motivated to learn. The team projects courses offer an excellent environment to develop new knowledge with activated good use of prior knowledge. In the teams, students benchmark their knowledge on a continuous basis, which synergistically activates prior knowledge in all team members. It does normally not need to have a diagnostic assessment of prior knowledge. As teams are to some degree self-directed, individual students tend to correct any knowledge deficiencies on their own. During the design team students have to develop requirements for the project, evaluate design architectures and find a system that they will manufacture and test. This effort requires a significant amount of brainstorming, which reveals prior knowledge in each student. The concepts they develop for the chosen architecture will include subsystems that are interfaced with energy or data flow, they may include mechanical, software and electronics components.

In an early phase students predict the performance of their project gadget based on select assumptions and perhaps back-of-the-envelope calculations. A test plan needs to be developed to verify the predictions. Teams have to justify why their design will work and how it will work. Errors and mistakes often happen at interfaces between subsystems. In the team discussions prior knowledge that has developed in the design process again will be activated in all team members with a good chance that most errors can be corrected. If not in the design phase errors will show up during implementation and operation phase of the project.

Project-based learning provide excellent vehicles to activate prior learning. Team working triggers prior knowledge in all team members. Projects are seen in the context of engineering applications which increases motivation and helps activate prior learning and increases the efficacy of the student. Incorrect prior knowledge will be corrected after personal initiative of the individual student, or at the initiative of classmates in team projects, thus relieving the teacher from having to carefully assess the prior knowledge of each individual student.

Principle 2: Knowledge Organization

University students tend to be presented with and learn largely disassociated "elements" of knowledge – facts, concepts and methods. Yet knowledge is not simply a set of disconnected facts, it is a *system* with facts, concepts and methods, as well as a network of connections among these elements, a noetic structure. In the language of engineering systems – knowledge is a *system*, with both entities (facts, etc.) and relationships (connections). The emergent property of this system is the ability of the thinker to solve problems and apply the knowledge in new ways.

To quote Ambrose: "As experts in our fields, we create and maintain, often unconsciously, a complex network that connects the important facts, concepts and procedures, and other elements of knowledge within our domain. Moreover, we organize our domain knowledge around meaningful features and abstract principles. In contrast, most of our students have not yet developed such connected or meaningful ways of organizing the information they encounter in our courses. Yet how they organize their knowledge has profound implications for their learning."

This line of reasoning leads to Ambrose's second principle:

> ***How students organize knowledge influences how they learn and apply what they know***.

Embracing this observation, we as experts can start to appreciate the subtlety of the organization of knowledge we possess. For example, an expert on solid mechanics would certainly hold a structure among the principles and methods to solve a mechanics problem – equilibrium, compatibility, and constitutive relations, for example. But the organization of knowledge even in our own minds is not unique – the same solid mechanic would have a structure around how equilibrium is used in a number of fields, and another on how certain tensor relationships interrelate.

We must then realize the lengthy development of this organization, and how much better we might perform as instructors if we explicitly help students develop the organization. As a simple example, we remember how confused we were by the relationships among force, work, energy and power. It was well into our university studies before this structure became intuitive, despite the fact we had been confronting it for almost four years. To take a more complex example in engineering, almost all struggle with the relationships between the second law of thermodynamics and other knowledge. It was only after Claude Shannon made the connection between work/heat and signal/noise that he was able to articulate the famous information theorem that now bears his name.

Ambrose cites a number of methods that instructors can use to make students more aware of the organization of knowledge, including having both the instructor and students draw concept maps, making explicit the organization of the course, and making connections between concepts explicit. These approaches are primarily focused on classroom instruction.

Alternatively, we can use projects as a learning medium for conveying structure. The execution of a design-implement project by a student allows them to understand the authentic organization of knowledge for synthesis. If executed early in the education, it can provide a foreshadowing of the details of the knowledge that will be learned, and give a roadmap for its organization. Projects can bring out deep organization of knowledge as a highly interconnected body, in contrast with courses that normally present a serial view of facts and methods. This is particularly true over the semesters of an engineering degree, where projects can scaffold skills progressively (we'll return to scaffolding later).

Consider an example of learning knowledge organization using the Lighter-Than-Air design-implement project for first year students developed at MIT. Students in teams of about six design a remotely piloted buoyant airship, driven by a electric motor driving a propeller [7]. They then build the airship, using balloons full of helium, lightweight structural materials (soda straws and tape), and a RC controlled servo controlling either a rudder or propeller. Inside a gymnasium, a competition takes place in which the students fly their airship around a set of pylons for minimum time.

This project is deliberately designed to be rich with knowledge relationships, and to be an authentic experience in aerospace design in which first year students can be successful. Among the many organizations of knowledge, ones the students encounter are:

- The relationships among stiffness, strength, weight, and structural density of lightweight structures
- How vehicle configuration, power, drag and maximum velocity are connected
- The equilibrium relationship among configuration volume, mass and lift
- The relationship between available electric power, thrust, delivered thrust power and propeller design
- How turn rate is interconnected with body shape, forward velocity, side force and control authority

Another process related organization of knowledge that the design build experience illustrate are the sequential (but often iterative) steps of requirement analysis, design, build and testing. This is a great example of an activity leveraging project-based and contextual learning.

The student will obviously not leave the first year course deeply understanding all of these individual topics (configuration, vehicle dynamics, propeller design, customer requirements, etc.) but they will develop a sense of the relationships among the topics. Contrast this resulting structure of knowledge that the student would develop with the traditional scenario in which a student learns structures in one track, fluid mechanics and applied aerodynamics in another track, propulsion in a third, and vehicle flight dynamics in a fourth, perhaps later integrated in a capstone design or design-build class.

In a corroborating exhortation, Harvard's David Perkins notes that effective teaching will seek to introduce the student to "play the whole game." [8] By this he means that learning goal's is the ability to DO something. The student sees the value of component knowledge in the context of their ability to perform some larger task. The challenge for teachers is the design of student work that's a whole activity, setting the context. One of the failures of reductionist engineering science is the descent into detail, remote from context, and the consequent disconnect from application, the student's perception that they're now enabled to do something with their knowledge.

Perkins observes, "We can ask ourselves when we begin to learn anything, do we engage some accessible version of the whole game both early and often? When we do, we get what we might call a 'threshold experience', a learning experience that gets us past our initial disorientation and into the game." In engineering education, that means getting students into the “whole game” of conceiving-designing-implementing-operating an engineering system in a mission or enterprise context. Furthermore, Perkins insists such experiences be done not simply as a capstone, but “early and often.”

The use of project-based learning as a way to develop structure is further supported by the Ambrose’s research summaries. The first of these points is that “no organizational structure is necessarily better or more ‘correct’ than another.” She goes on to point out that knowledge organizations are most effective when they are well matched to the way that knowledge will be accessed or used. From this we can see that the desirable organizational structure for students who will go on to engineer – that is design and build new systems – is in fact the structure that would evolve by practicing designing and building in projects while students. This is in contrast, perhaps, with the organization that is understood by a faculty member who is a researcher in the field. So project-based learning with authentic engineering activities will develop the organization that will support successful engineering development.

When students are exposed to an organization of knowledge before being exposed to the details, they are actually better able to learn. Ambrose states “student show greater learning gains when they are given an *advanced organizer*, that is, a set of principles or propositions that provide a cognitive structure to guide in the incorporation of new information.” This is exactly one of the roles of an early design-implement project– to build the cognitive framework for the future learning of technical fundamentals. When faculty who teach theory question the value of “wasting time” on early projects, this is the most effective argument – that the projects will help the students learn the abstractions that follow. Early exposure “whole game” provides context for the necessary disciplinary detail to follow.

A third point that emerges from the research is that novices tend to develop more simplistic or superficial organizations of knowledge than do experts. Experts recognize meaningful patterns. Engineering students may, for example, classify equations as linear, quadratic, or differential, while a more expert observer may classify them by the phenomena the equations represent. Ambrose concludes that “we need to provide students with the appropriate organizing schemes or teach them how to abstract the relevant principles form what they are learning.” One way to teach deeper organizing schemes is to use project, and particularly advanced design-build projects in the later years of education.

In summary, design-implement projects are an excellent way to apply the principle of knowledge organization. Projects build an authentic organization that will be useful for students’ future, in the earlier years of education, they can be used for advanced organizers, and in the later years they serve to teach deep, and not superficial, organization.

Principle 5: Practice

Project-based learning should positively implement Ambrose’s fifth principle:

goal-directed practice coupled with targeted feedback are critical to learning.

In order to be successful, conceive-design-implement projects must have clearly defined goals for the students to achieve, and students should have multiple opportunities to practice such projects within their undergraduate curriculum (e.g., “cornerstone” and “capstone” design projects). Conceive-design-implement projects inherently have multiple opportunities for targeted feedback on the project itself, as many projects are managed using a stage-gate or phased approach to ensure timely completion. Such project can also be paired with writing-intensive course requirements to provide additional opportunities for feedback—and practice—beyond the project itself.

We'll use Penn State's Learning Factory as an illustration of the application of this methodology. At Penn State, Ambrose's fifth principle is best embodied within the capstone design projects undertaken by senior engineering students. Like many universities, these capstone design projects are performed collaboratively with industry, allowing for multiple levels of feedback. When submitting their project descriptions, companies are explicitly asked to clearly define the goals and deliverables for their project, which each instructor reviews prior to the start of the semester and then with the students when classes start. This ensures that not only are the students and faculty on the same page, but both parties are on the same page with the company sponsor as well. Ambrose warns that "Instructors often think they are conveying specific goals to students when, in fact, they are not", and this goal setting practice occurs on multiple levels to ensure a successful capstone experience. For instance, students and sponsors are encouraged to sign off on a Deliverables Agreement during their initial site visit to the company after they have carefully reviewed the project with the sponsor. This helps ensure that "goals are stated in such a way that students' performance can be monitored and measured", while ensuring student "buy-in" with the project. Finally, given the breadth of ABET criteria that the capstone course satisfies in most departments, course syllabi in capstone design courses across the College of Engineering at Penn State are carefully reviewed each year and agreed upon by faculty in other departments to allow students to move seamlessly between sections. This allows single and multi-disciplinary design project teams to be easily formed based on the needs of the projects as each department has already agreed upon what it "really wants students to learn" through this syllabus review process. As the capstone design projects are executed, they naturally satisfy the dual goals of the sponsoring companies and the faculty.

Ambrose cautions that "practice [students] do should be at an appropriate level of challenge and, as necessary, accompanied by appropriate amount and type of support". While some faculty utilize some form of knowledge and skill assessment at the start of their capstone design course as Ambrose suggests, students at many institutions self-select projects on which they would like to work for the semester or year. In particular, capstone design project descriptions across the participating departments are compiled into a single list, which is distributed to all faculty and students (each Fall semester, 8 engineering departments participate in the capstone design program coordinated by Penn State's Learning Factory; in Spring, this number jumps to 11). Students are allowed to work on any project that needs their disciplinary expertise (e.g., an industrial engineering student can work on any project that needs an industrial engineer but not a project that needs, for example, only electrical engineers). During the first week of the semester, all of the company sponsors participate in a Project Kickoff even wherein they field questions from students that further inform their project preferences. A student typically comes to the Project Kickoff with 6-10 projects in mind and then uses the Kickoff to identify which projects are of most interest and at the right level (e.g., scale and scope) to challenge them. At the end of the Kickoff, students rank order their top five project preferences, and this information is then used to assign students to teams (3-5 students per team), balancing course and project needs with student interests and skills as best as possible. This process allows students to match their skills and interest to an appropriate level of challenge, helping to avoid the pitfall that Ambrose notes when there is a mismatch: "If a challenge is too great, learners may have a negative expectation for success and hence become disengaged and apathetic. In contrast, if students feel that the challenge is reasonable, they will likely hold a positive expectation for success that will increase their tendency to persevere and work hard for the goal."

Support for each project then occurs on two levels. First, faculty meet regularly with each team to review their progress. Each team is required to prepare a weekly memo indicating what they have accomplished in the previous work, what they are working on during the current week, and what they plan to work on in the coming weeks. This weekly memo is also sent to the company sponsor, who provides technical support for the project. This weekly interaction with industry is what makes the capstone design projects so successful—without this “real world” element and the interactions with subject matter experts from industry, these projects would be no different than any other class project. Mid-semester and end-of-semester evaluations from the sponsors are also used to provide guidance to the teams, and most faculty include these evaluations as a portion of each team’s grade (e.g., 5%-10%). Finally, teams are also encouraged to seek additional faculty or staff support as needed for a project to reinforce the importance of self-directed learning in the course.

These weekly meetings help ensure that “Goal-directed practice [is] coordinated with targeted feedback in order to promote the greatest learning gains.” Faculty also meet with all of their teams as a class once a week to provide feedback at the group level, e.g., project reporting requirements, expectations for presentations, etc. Project reports are tied closely to the stages of the project (e.g., concept development, preliminary design, detailed design, final design), and a report template and example reports are provided to the students to “show students examples of what the target performance looks like”. Coupling feedback with practice, faculty also require draft versions of team reports 2-3 times during the semester. This allows team-level feedback and the opportunity to revise the report (usually within a week’s timeframe) and improve their technical writing skills through repeated practice. Students are also encouraged to review past project posters (on display throughout the hallways in participating departments) and project summaries (available online through the Learning Factory’s website) and prepare drafts for faculty to review prior to the end of the semester when they are due. To facilitate this process, most faculty provide students with a copy of the grading rubric used for each report. As Ambrose points out, “This [helps] students become better at identifying the qualities of good work and diagnosing their own problems.” Likewise, the judging criteria that are used by industry judges to identify the Best Projects and Best Posters at the Design Showcase are shared with students early in the process to reinforce the course goals and sponsor interactions.

Finally, peer-review has become a tertiary, yet important, element of support and feedback during the semester-long project. During in-class presentations, students are asked to grade their peers using the same rubrics that the instructors use, and copies are provided prior to the presentations so that students have specific goals to direct their attention. This allows them to “provide constructive feedback on each other’s work”, and peer evaluation is used periodically throughout the semester as part of a “team check-up” to identify problems with team dynamics. The timing and level of this feedback is critical—done too soon, it cannot be used effectively and is seen as “busy work”; done too late, and there might not be enough time to address the issue—and we continue to fine-tune the process each year to maximize its effect.

Principle 7: Metacognition

Metacognition speaks to the ability to step back from the immediate task and evaluate one’s own thinking and learning. Bransford defines *metacognition as* “The process of reflecting on and directing one’s own thinking.”[9] Importantly, this is a distinct skill requiring purposeful development. Ambrose’s principle:

> **to become self-directed learners, students' must learn to assess the demands of the task, evaluate their own knowledge and skills, plan their approach, monitor their progress, and adjust their strategies as needed**.

Each of these five noted skills is indispensable for self-awareness and self-direction in learning, and indispensable in the modern engineering environment. According to Ambrose, the research significantly notes, “Students who were taught or prompted to monitor their own understanding or to explain to themselves what they were learning had greater learning gains relative to students who were not given any monitoring instruction.”

Design-implement projects are time intensive, and class hours devoted to the messy phases of project work are hours not devoted to new content. “I don’t have time for projects” is a common obstacle to their adoption. Yet, it’s in that messy project context that students must muddle through their understanding of the task, confront gaps or misconceptions in their knowledge, strategize approaches to back-fill those gaps or apprehend new material, defend their progress to teammates, and re-adjust strategies as deadlines loom and pressures build. Principle 1 above spoke to prior learning and the dissonance of faculty expectations and student presumption. In a project context, students confront their misconception or shallow learning.

Ambrose’s general prescription: “students will often need our support in learning, refining, and effectively applying basic metacognitive skills. To address these needs then requires us as instructors to consider the advantages these skills can offer students in the long run and then, as appropriate, to make the development of metacognitive skills part of our course goals.” This goal fits neatly with project-based learning and reflects learning outcomes explicitly found in the CDIO syllabus (particularly 2.1- Analytical Reasoning and Problem Solving, 2.2- Experimentation and Knowledge Discovery, and 2.3- Personal Skills and Attributes).[2]

Assessing the Task at Hand

Students commonly enter engineering assuming that problem solving in engineering will resemble the closed-form problem solving they performed in pre-requisite science classes. “Given XX, find YY.” The transition from competent science student to competent engineer will entail weaning them, and progressively setting before them tasks with increasing ambiguity, and multiple solution paths. Early courses can provide more structured tasking and deliverables, whereas upper-class courses could deliberately refrain from explicit detail. The charge would be that students recognize that they must answer specific detailed questions, but the senior should be able to thoughtfully compose the questions they’re to either answer themselves, or draw from their project sponsor or client in negotiation. A sophomore project might explicitly list the questions to be answered; a junior project might flesh out the tasking as a class exercise; a senior project would leave evident questions unstated, expecting the student or their teams to ascertain the detail.

This process Ambrose describes as *scaffolding*, and appears to suggest its use in the context of individual courses. However, scaffolding is particularly well suited to a program perspective on such tasks a technical writing, teamwork, and experimental studies, and should be regarded as a substantial tool for addressing CDIO Standard 3- “Integrated Curriculum.”[2] Scaffolding may successfully be applied to many of our metacognitive skills. For example, the Naval Academy’s aerospace program embeds technical writing instruction in a series of disciplinary courses where the three-year progress begins with structured technical writing assignments through more advanced assignments with less explicit structure, and culminating in senior-level writing assignments with deliberately vague guidance at points. The student is compelled to stand on their own, evaluating the audience and message, and tailoring the writing product accordingly. Moreover, they’re told that the guidance is deliberately vague, and that they’re being conferred responsibility for determining the appropriate scope and detail for their writing products. Hence, the scaffolding is progressively removed over six semesters’ time.

Rubrics provide another means by which students can be boosted towards better assessment of the task, as well as the later task of monitoring their own progress. A variety of CDIO collaborators have been involved in the development and publication of rubrics suitable to project based learning.[10] This also supports and facilitates the targeted feedback described above in Principle 2.

Evaluating Strengths and Weaknesses

The tendency of students to overestimate their understanding is well-documented. The dissonance found in Principle 1 above frequently emerges from a student believing that knowledge at level 2 on Bloom's taxonomy represents mastery, whereas their professor, expecting level 4, regards them as clueless. For the maturity level we seek from a graduating engineer, they need not only formative assessment, but a means by which they can assess their own knowledge mastery. Rubrics, mentioned above, can address this purpose, as can deliberate programmed peer-review of student work. Practice in evaluating the quality of their peers, can be expected to promote better self-reflection. Healthy teams will naturally engage in self-reflective processes gauging one another's work. Those whose work efforts are more polished will commonly place pressure on other team-members to bring their work up to team norms, or lagging team-members will be self-conscious of the disparity in their presented work in team presentations. That self-consciousness of the quality of a student's own work is exactly what we want to promote.

Planning an Approach

An interesting feature of the expert-novice studies reviewed by Ambrose (such as [9]) includes the disparity in time allocated by experts to planning their work. Experts solve problems faster than novices, but spend considerably more time proportionally to their planning of the approach. Students need explicit instruction in how to plan their work. Those who've taught computer programming to undergraduates will have seen this in the habit of students to jump right into typing code without having thought about the structure of the data that they'll need at the end of the problem, or the natural breaks in the program, or the loci at which they might assess the program's accuracy. On larger scales, project planning is explicitly found in the CDIO syllabus within objective 4.3- "Conceiving and Engineering Systems."

Ambrose specifically suggests three practices commonly found in larger-scale projects such as capstones. Her suggestions are themselves scaffolded, to expose the student to the process of planning their work:

- have students implement a provided plan
- have students create their own plan for their work
- make planning the central goal of an assignment

Here's a point at which many project oriented programs could improve, as project planning elements are commonly described in the context of capstone projects in the senior year, but likely not treated previously. The sequence above lends itself to progressive complexity year-to-year in a majors program, where planning skills were explicitly introduced, taught, and utilized over a multi-year program, much as described above for technical writing. Few engineering enterprises succeed for long absent good enterprise and project planning. This is a validated professional skill, which needs nurtured in undergraduates, and valued by faculty. [11]

Applying Strategies for Monitoring Performance

A critical feature of learning to plan their work is that the plan becomes a yardstick for measuring performance. Industry understands this. Budgets are built and real spending compared to projected. Cost and schedule variance are key management measures in most engineering enterprises. Students need to see the value of even the simplest project plans in monitoring and adjusting their work. Rubrics and peer-review, as discussed above, are vital tools and attributes of project learning, particularly those involving teams. Additional tools addressing this challenge can be found in Koster. [12]

Reflecting on and Adjusting One's Approach

In this section, Ambrose's direct suggestions are contextualized for a traditional lecture course with exams for summative assessment. Yet, the value of this skill is no less applicable to the design-implement experience. We'll highlight two.

"Students should be led through activities that require reflection on their performance." An error we've all committed is packing a short duration project at the end of the term, and allowing a report, competition, or presentation to close the semester. A substantial learning opportunity is forfeited if students individually and as teams are not compelled to reflect on both the quality of their work, *and* all the processes that got them there, to include technical and organizational factors. In the midst of a design team's dysfunction, the pressure to deliver the report on time masks the lesson to be learned about team behaviour. Only absent the deadline can the team more dispassionately glean what they're to learn about the process of making teams work. Early design decisions may have committed them to paths that closed viable alternatives. Those alternatives could now be contemplated for their strengths and weaknesses, in retrospection. In year-long design courses, this activity occurs more naturally provoked by faculty submitting grades for the first semester. For example, a preliminary design review followed by a critical design review allows students to change designs with a slightly different system to achieve the same outcome but more efficiently or at lower cost and labor.

"Create assignments that focus on strategizing rather than implementation." This offers intriguing opportunity for the engineering educator. One particular modern challenge is getting students (and some faculty) to develop *systems perspectives* on engineering design. The systems engineering challenge is particularly keen at the interfaces where a design critically interacts with other systems over which the design team might have no control (the internet, the air traffic control system, GPS, etc.). Implementation projects (Design-build-operate) must necessarily fit within the scope of the time and resources of the academic year and the campus infrastructure (people and facilities). Conceive projects however, short of detailed design, can be of tremendous breadth, while limited scope, with an analysis of alternatives, and can be scaled to fit the time, while fostering the development of a broader systems perspective. Examples might include risk reduction approaches for a vehicle test, analysis of alternatives for landing on the moon, or the air-defense system for a nation's capital. These strategy designs can focus on the Conceive phase, permitting students to more immediately see their ultimate need to understand the societal or business contexts in which they'll ultimately work.

In her 2007 plenary address to the CDIO conference at MIT, Susan Ambrose explained that weak metacognition was characteristic of this particular generation of students.[13] Project-based learning appears an effective medium for tackling this thorny goal of education. John Henry Newman in *The Idea of the University* keyed on metacognition as the very point of the liberal education, "The man of developed [mental] faculties has command of others' knowledge. The man without them, commands not his own."[14] If that was a true and desired aim of education in 1852, how much more so now?

The Other 3 Principles

We've neglected three of Ambrose's principles not because they do not pertain, but rather because we felt the four above most deserved discussion. Of the three that remain, the first two connect with project-based learning so strongly as to be almost self-evident.

- **students' motivation generates, directs, and sustains what they do to learn.**

Sceptics of project-based learning commonly assume that student motivation is the dominant reason for other faculty members' enthusiasm. Motivated students certainly motivate faculty. Our goal here has been to substantiate the pedagogical foundation for project-based learning as far more profound than simply making engineering fun for students. Though, we won't be embarrassed to enjoy ourselves when students have fun while learning.

- **to develop mastery, students must acquire component skills, practice integrating them, and know when to apply what they've learned.**

The principle of Mastery virtually screams for project-based learning, particularly that spanning multiple semesters such as found in the integrated curriculum sought by CDIO programs (CDIO Standard #3).

- **Students' current level of development interacts with the social, emotional and intellectual climate of the course to impact learning.**

This last may deserve a future paper of its own, as we believe project-based learning leverages the multi-dimensional development of the undergraduate, fostering much more than their intellectual development.

CONCLUSIONS

The CDIO consortium has struggled to substantiate our zeal for project-based learning with research quality evidence that our pedagogy works better than that which we seek to reform. Susan Ambrose and her colleagues have provided those interested in improving teaching with substantiation and instantiation of means by which the learning sciences can be brought directly to bear on college teaching and learning. We've briefly considered their distillation as it applies to the project-based learning as embraced by CDIO programs. Our foundational hope is that our peers would consider their work seriously, and join with us in working out together the implications for our programs and our teaching.

REFERENCES

[1] Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010). *How learning works: Seven research-based principles for smart teaching*. San Francisco, CA: Jossey-Bass.

[2] Crawley, E. F., Malmqvist, J., Östlund, S., & Brodeur, D. R. (2007). *Rethinking engineering education: The CDIO approach*. New York: Springer.

[3] Wilkerson, L., & Gijselaers, W. H. (Eds.). (1996). *Bringing Problem-Based Learning To Higher Education: Theory And Practice.* New Directions for Teaching and Learning, No. 68. San Francisco, CA: Jossey-Bass.

[4] Bransford, J. D., & Johnson, M. K. (1972). Contextual prerequisites for understanding: Some investigations of comprehension and recall. *Journal of Verbal Learning and Verbal Behavior, 11*, 717-726.

[5] Mazur, Eric (1996). *Peer Instruction: a Users Manual* Benjamin Cummings.

[6] Clement, J. J. (1982). Students' preconceptions in introductory mechanics. *American Journal of Physics*, *50*, 66-71

[7] Crawley, E.F., R. J. Niewoehner, Jean Koster. "North American Aerospace Project: Adaptable Design/Build Projects for Aerospace Education," *AIAA Aerospace Sciences Meeting*, Orlando, FL, Jan, 2011.

[8] David N Perkins (2009) *Making Learning Whole: How Seven Principles of Teaching can Transform Education*. Jossey Bass, San Francisco.

[9] Bransford, John D., Brown, Ann L., and Cocking, Rodney R. (editors), *How People Learn: Brain, Mind, Experience, and School* (expanded edition), National Academy Press, Washington, D.C., 2000.

[10] Daryl G. Boden and Peter J. Gray (2007), "Using Rubrics To Assess The Development Of CDIO Syllabus Personal And Professional Skills And Attributes At The 2.X.X Level," *Proceedings of the 3rd International CDIO Conference,* MIT, Cambridge, Massachusetts, USA.

[11] Honor J. Passow "What Competencies Should Engineering Programs Emphasize? A Meta-Analysis of Practitioners' Opinions Informs Curricular Design." *Proceedings of the 3rd International CDIO Conference,* MIT, Cambridge, Massachusetts, USA.

[12] Koster, Jean N. "Challenges in Grading Project Teams." *Proceedings of the 6th International CDIO Conference*, École Polytechnique, Montréal, June 15-18, 2010

[13] Ambrose, Susan. Plenary address to the 3rd International CDIO Conference, MIT, Cambridge, Massachusetts, USA.

[12] Newman, John Henry (1852).*The Idea of the University.* Notre Dame Press, Notre Dame, IN.

Biographical Information

Robert J Niewoehner is the David F. Rogers Professor of Aeronautics at the United States Naval Academy. Dr. Niewoehner entered academia after a career as a US Navy pilot and experimental test pilot, including service as the US Government's Chief Test Pilot for the development of the F/A-18 E/F Super Hornet. He initiated the Naval Academy's involvement in CDIO in 2002 and has been an active collaborator since. His scholarly interests include flight mechanics, flight test engineering, engineering education reform, and critical thinking in higher education.

Edward F. Crawley is Ford Professor of Engineering and Professor of Aeronautics and Astronautics and of Engineering Systems at MIT. He is Co-Director of the CDIO Initiative, and co-author of *Rethinking Engineering Education: The CDIO Approach*. He currently directs the MIT-Bernard M. Gordon Engineering Leadership, and was awarded the 2011 Gordon Prize for Engineering Education by the National Academy of Engineers.

Jean N. Koster is Professor of Aerospace Engineering Sciences at the University of Colorado, Boulder, Colorado and President of Tigon EnerTec, Inc., a start-up company for aerospace propulsion technologies. He is the CDIO representative from the University of Colorado and is the department course coordinator for the senior design projects courses.

Timothy W. Simpson is a Professor of Mechanical & Industrial Engineering at Penn State with affiliations in Engineering Design and the College of Information Sciences and Technology. As the Director of the Learning Factory, he coordinates over 120 industry-sponsored senior capstone design projects each year for nearly 600 students in the College of Engineering. His research spans several areas in engineering design, including product family and product platform design and multidisciplinary design optimization.

Corresponding author
Dr. Robert J Niewoehner
590 Holloway Rd.
Annapolis, MD 21409
1-410-293-6402
niewoehn@usna.edu