# STUDENT PERCEPTIONS OF PROJECT-BASED LEARNING IN A SOFTWARE ENGINEERING COURSE

**Dina Shoham**

McGill University, Montréal, Canada

**Robyn Paul, Mohammad Moshirpour**

Schulich School of Engineering, University of Calgary, Calgary, Canada

## ABSTRACT

This paper presents an overview of a second-year programming course in the department of software engineering at the University of Calgary. The course was recently restructured to include aspects of project-based learning (PBL) to help students meet graduate attributes and practice learning outcomes that a traditionally formatted course may not allow them to achieve. This restructuring centered around the implementation of a final term project that students completed in the final three weeks of the course. While this format does not necessarily follow the typical PBL approach, where a project is typically conducted throughout an entire semester while simultaneously acquiring disciplinary knowledge (often in other courses), it offers instructors a more accessible approach to PBL implementation that does not require restructuring at the department or faculty level. The project introduced to the course closely resembled a genuine industry project, and thus allowed students to experience what the software industry can be like, providing them with valuable experience. Data was collected in the form of a Likert-style survey that many of the students completed and supplemented with a descriptive questionnaire to which both the professor and handful students responded. This data was then analyzed using a theoretical framework based on relevant CDIO standards, and relevant findings are discussed alongside areas for improvement and further research. Students' response was generally quite positive, and the professor observed they benefited quite significantly from the implementation of PBL in the course.

## KEYWORDS

Project-based learning, Software engineering, Hybrid PBL, Student perceptions, Standards 1, 2, 5, 7, 8

## INTRODUCTION

### *The need for change in (software) engineering education*

The field of engineering is ever-changing and, with it, the pressure on post-secondary institutions to develop and maintain a curriculum that can keep pace. Countless studies show that Canadian engineering graduates are not equipped with the skills and knowledge the industry demands of them (May & Strong, 2011). Similarly, graduate attributes published by different engineering accreditation boards are becoming more demanding. Specifically, the Canadian Engineering Accreditation Board expects students to not only have exceptional disciplinary knowledge and skills but also to develop their interpersonal skills, and be able to situate their work in the broader societal context in which they operate and to be lifelong learners ("CEAB Graduate Attributes, n.d.). Having a holistic skillset is incredibly important for engineers graduating today. In its Final Year Engineering Students 2017 Survey, with responses from 2,485 graduates from institutions across Canada, Engineers Canada found many students felt unprepared for or disconnected from the engineering industry, especially with respect to their coursework (most of the positive responses on the survey relating to industry preparedness was linked to students who completed work terms such as co-ops or internships) ("Final Year Engineering Students 2017 Survey - National Results", n.d.).

According to CDIO, the central issue in the problem of the discrepancy between engineering students' capabilities and industry demands is the tension between the need to develop students' technical knowledge with personal, interpersonal, and product, process, and system-building skills (Crawley, Malmqvist, Östlund, Brodeur, & Edström, 2014). While much of the engineering curriculum focuses on technical knowledge, these other skills are often overlooked, especially within the context of the traditional lecture learning format.

Software engineering, in particular, is a relatively new field compared to other engineering disciplines and is characterized by consistent, rapid technological development (Mead, 2009). As such, the demand for software engineering educators to adequately prepare their students for the industry is significant, and often software engineering programs fail to meet this demand (Shaw, 2000).

### *A potential solution: project-based learning*

While resolving the rift between student capabilities and industry demands is a vast and complex issue, one solution many engineering educators are turning to is project-based learning (hereafter referred to as PBL). Countless case studies - several of these are reviewed in the following section - show the implementation of some form of PBL to be highly beneficial to students' learning, and encourages the development of many of the crucial skills that CDIO highlights. A review of research on active learning notes "extensive and credible evidence suggests that faculty consider a non-traditional model for promoting academic achievement and positive student attitudes." (Prince, 2004). PBL is an effective example of this kind of non-traditional learning methods.

Projects in software engineering can, when properly executed, provide an opportunity for students to practice the process of conceiving, designing, implementing, and operating an industry-relevant project, and help to facilitate the development of key skills and attributes (Crawley et al., 2014).

This paper will first summarize some of the relevant literature within the domains of project-based learning and (software) engineering education. It will then explain the methodology used to analyze student data, which involved selecting relevant CDIO standards and using them to examine a student survey and some comments. Interesting findings are then discussed, and finally, suggestions for improvements and future research are outlined.

## BACKGROUND

There are countless instances of project-based learning being applied in engineering education. The majority of the literature suggests that PBL is both an effective and necessary strategy to engage engineering students, as well as to provide them with the opportunity to have a closer learning experience to what will actually be required of them in the future. In a twenty-year study conducted in Spain, professors found that by employing PBL, undergraduate engineering students consistently took greater responsibility for their own learning, actively immersed themselves in meaningful projects, and especially developed personal competencies including teamwork, motivation to learn, and creative problem solving (Ríos, Cazorla, Díaz-Puente, & Yagüe, 2010). This finding is reaffirmed by a study conducted at Massey University, which concluded that the implementation of PBL in their engineering courses resulted in increased student motivation, improved problem-solving skills, and a better understanding of the engineering design process (Shekar, 2014). Another study at the University of Adelaide of a final year honors project for mechanical engineering students found "student engagement [...] increased only after the amalgamation made clear the links to authentic engineering practice." (Prime, Robertson, Cazzolato, Missingham, & Kestell, 2015). Another study done on a capstone project at Carleton University's Department of Systems and Computer Engineering came to a similar conclusion: that students were able to complete a project that closely resembled industry standards (Schramm & Chan, 2013). All of this points to the fact that PBL is an extremely effective strategy in bridging the gap between students' formal education and industry – this is crucial in such a career-oriented field of study.

Much of the research of PBL and alternate learning approaches has become even more relevant in the field of software engineering and computer science since it is a uniquely demanding field in terms of teaching strategies. A review of computer science projects found that in a PBL-based course, students were able to attain and apply disciplinary knowledge and skills while simultaneously developing teamwork and project management skills (Pucher & Lehner, 2011). Even when students received lower grades on projects in project-based courses, instructors determined they had attained higher learning outcomes. A similar case study at Oslo and Akershus University College of Applied Sciences found that so long as learning expectations were clear, student motivation was boosted by the inclusion of PBL in a programming course (Zouganeli, Tyssø, Feng, Arnesen, & Kapetanovic, 2014).

Essentially, in both general engineering education and specifically in software engineering education, PBL is an extremely effective method for getting students to apply their disciplinary knowledge, and at the same time, develop many of the disciplinary and interpersonal skills they will need in industry. This multifaceted approach has been a successful strategy for many educators in order to meet the ever-increasing demands for engineering graduates.

However, the typical PBL approach can be extremely demanding for instructors. (Stoicoiu & Cain, 2015) PBL restructuring often requires coordination with other courses so that students can learn the disciplinary knowledge required of them for their project in time. This can even require faculty-level reorganization, which is simply not viable for many instructors. As an alternative, instructors can opt for a hybrid PBL approach, in which the earlier section of the course is taught more traditionally (using lectures, assignments, and labs), and only towards the end of the course are students given a PBL-style project. This project would have to be less extensive in scope, but can still result in many of the positive outcomes of the typical PBL approach, while also being accessible and practical at the individual course level.

### *ENSF 409: a case study in PBL in software engineering education*

Software Engineering for Engineers (ENSF 409) is a core course for second-year students in software engineering at the University of Calgary. It covers topics including object-oriented programming and application of data structures and strategies for testing and debugging. In past years, the course was taught using a very traditional format, relying heavily on lectures and limited-scope assignments. However, the course was recently restructured to include a term project, modeled after the PBL paradigm, and introducing a new focus on software and coding best practices.

The term project is worth 10% of students' final grade and involves the development of a client-server application. It is completed towards the end of the semester and implements some aspects from previous assignments. The students are given the option to complete the project individually or in groups of up to three and the project is completed in four stages: a pre-project exercise that permits students to familiarize themselves with some programming strategies they will be employing later on, a design submission, and two implementation demonstrations.

Students are provided with the problem statement, but no other directions, in order to simulate an industry project, as well as to encourage them to research and design a solution without much guidance. The open-ended nature of the project also worked to encourage students' creativity, while still requiring them to apply their knowledge of object-oriented programming, client-server architecture, and software engineering best practices. Past iterations of the project have included an online learning platform for students and teachers to interact, a tool shop application, a course registration platform, and more.

## METHODOLOGY

### *The Data*

The data used to analyze student perceptions on the effectiveness of the recent implementation of PBL in ENSF 409 includes a student survey, which had a 33% response rate (43 out of 130). The survey included 30 questions on a 5-point Likert scale that asked students about the level of effort, course contributions to learning, course delivery, and content, and questions about the term project specifically. An additional questionnaire was also sent out to a handful of students to gather more specific comments on certain aspects of the project. Finally, the course instructor was also given a brief questionnaire, again used to gather additional details on the implementation and design of the project. Overall, the quantitative data from the in-class survey, combined with the qualitative data from the follow-up questionnaires, provided a rich source of data for analysis.

### *Theoretical Framework*

To analyze students' data in a methodical fashion, a theoretical framework was used for the data analysis (Grant & Osanloo, 2014). The framework included five of the twelve CDIO standards. Because this paper only covers data from a single, core course, and due to the nature of the course itself, many of the CDIO standards were not relevant to our analysis. The rationales for the CDIO standards chosen are summarized in Table 1. We selected the CDIO standards as a means by which student success can be evaluated since the CDIO standards and syllabus are a direct response to the increasingly demanding expectations for graduating engineering students; research shows that experimentally, engineering programs modeled on a subset of the CDIO syllabus will likely achieve many, if not all of the CEAB graduate attributes (Cloutier, Hugo, & Sellens, 2012).

Table 1. CDIO Standards selected for use in the theoretical framework

| ***Standard*** | ***Reason for Inclusion*** |
|---|---|
| Standard 1: The Context | The process of development, testing, and demonstrations of the project design align with the conceive-design-implement-operate model, as well as the product-process-system lifecycle. Thus, the term project is situated within the CDIO context. |
| Standard 2: Learning Outcomes | The majority of the learning outcomes set forth in Standard 2 can be found somewhere in the outcomes pursued in the project, and provide a useful framework with which the benefits of the project to students can be identified and evaluated. |
| Standard 5: Design-Implement Experiences* | The term project is an example of a design-implement experience for second-year software engineering students, and so the CDIO guidelines for a design-implement experience provide a helpful guide. |
| Standard 7: Integrated Learning Experiences* | Standard 7 can be seen as a method by which many of the learning outcomes from Standard 2 can be pursued, and will be treated as such in this paper. |
| Standard 8: Active Learning | As with Standard 5, the project acts as an example of an active learning experience, providing students with an engaging and self-guided opportunity to apply their skills and knowledge. |

The five standards were mapped to relevant questions from the survey, as well as to questionnaire responses, in order to form a cohesive picture of the project's strengths and weaknesses within the context of the CDIO standards.

## RESULTS

### *Survey Data*

Table 2 shows each relevant question from the student survey and its average survey score. The survey followed a Likert scale, with 1-5 corresponding respectively to poor, fair, satisfactory, very good, and excellent for the first two categories, and to strongly disagree, disagree, neutral, agree, and strongly agree for the remaining four categories. The average response to all questions was 4.21, which we deemed a very positive overall response.

Table 2. Selected survey questions and corresponding CDIO standards

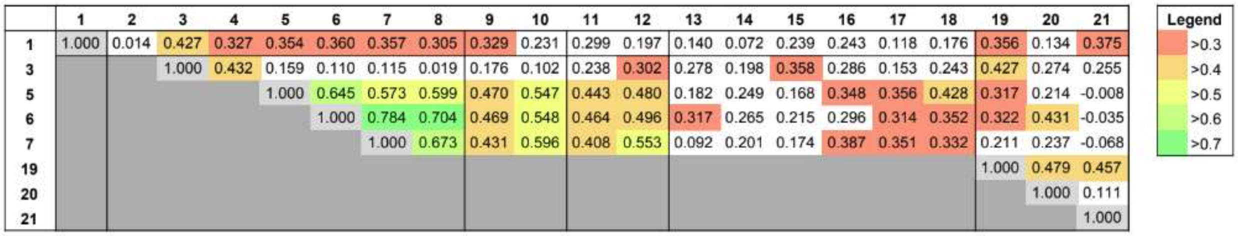
| **Category** | **#** | **Question** | **CDIO Std.** | **Avg. Score** |
|---|---|---|---|---|
| Level Of Effort | 1 | Level of effort you put into the course | * | 4.33 |
| Contributions To Learning | 2 | Level of programming skill/knowledge at the start of course | 2 | 3.14 |
| | 3 | Level of programming skill/knowledge at the end of course | 2, * | 4.33 |
| | 4 | Level of programming skill/knowledge required to complete the course | 2 | 3.67 |
| | 5 | Contribution to your skill/knowledge of programming | 2, * | 4.27 |
| | 6 | Contribution to your understanding of best coding practices | 1, 2, * | 4.21 |
| | 7 | Contribution to your understanding of software engineering best practices | 1, 2, * | 4.23 |
| | 8 | Contribution of course to your understanding of object-oriented design | 2 | 4.47 |
| Course Delivery | 9 | Instructor stimulated student interest | 7, 8 | 4.54 |
| | 10 | Lectures effectively prepared me for assignments and the final project. | 2, 7 | 4.24 |
| Course Content | 11 | Learning objectives were clear | 2 | 4.44 |
| | 12 | Course organized to allow all students to participate fully | 2, 8 | 4.49 |
| Term Project | 13 | My understanding of technical concepts increased | 2, 5, 7, 8 | 4.44 |
| | 14 | My creative thinking was improved | 1, 2, 5, 7, 8 | 4.24 |
| | 15 | My interest in programming increased | 1, 5, 7, 8 | 4.21 |
| | 16 | Lectures effectively prepared me for the final project | 2, 5, 7, 8 | 4.19 |
| | 17 | I learn industry-relevant skills by completing the project | 1, 2, 5, 7, 8 | 4.21 |
| | 18 | Assignments and projects helped prepare me for the software industry | 1, 2, 5, 7, 8 | 4.14 |

| | | | | |
|---|---|---|---|---|
| Student Perceptions | 19 | I enjoy computer programming | * | 4.72 |
| | 20 | I am interested in a career in software engineering | * | 4.53 |
| | 21 | I feel confident as a programmer | * | 4.02 |

The use of averages above assumes Likert-scale data values are continuous - although this assumption is not widely accepted in descriptive statistics, we determined the use of average values helped to inform our conclusions and thus decided to include them.

In order to determine correlations between responses to each question on the survey, the Kendall Tau-B test was used since our data is ordinal and non-parametric (Cohen et al., 2018). Figure 1 shows the values of the correlations. Tau values greater than 0.3 are considered statistically significant (Puka, 2011).

Figure 1. Kendall Tau correlation values for questions that indicate general success



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.000 | 0.014 | 0.427 | 0.327 | 0.354 | 0.360 | 0.357 | 0.305 | 0.329 | 0.231 | 0.299 | 0.197 | 0.140 | 0.072 | 0.239 | 0.243 | 0.118 | 0.176 | 0.356 | 0.134 | 0.375 |
| 3 | | | 1.000 | 0.432 | 0.159 | 0.110 | 0.115 | 0.019 | 0.176 | 0.102 | 0.238 | 0.302 | 0.278 | 0.198 | 0.358 | 0.286 | 0.153 | 0.243 | 0.427 | 0.274 | 0.255 |
| 5 | | | | | 1.000 | 0.645 | 0.573 | 0.599 | 0.470 | 0.547 | 0.443 | 0.480 | 0.182 | 0.249 | 0.168 | 0.348 | 0.356 | 0.428 | 0.317 | 0.214 | -0.008 |
| 6 | | | | | | 1.000 | 0.784 | 0.704 | 0.469 | 0.548 | 0.464 | 0.496 | 0.317 | 0.265 | 0.215 | 0.296 | 0.314 | 0.352 | 0.322 | 0.431 | -0.035 |
| 7 | | | | | | | 1.000 | 0.673 | 0.431 | 0.596 | 0.408 | 0.553 | 0.092 | 0.201 | 0.174 | 0.387 | 0.351 | 0.332 | 0.211 | 0.237 | -0.068 |
| 19 | | | | | | | | | | | | | | | | | | | 1.000 | 0.479 | 0.457 |
| 20 | | | | | | | | | | | | | | | | | | | | 1.000 | 0.111 |
| 21 | | | | | | | | | | | | | | | | | | | | | 1.000 |

| Legend | |
|---|---|
| 🟥 | >0.3 |
| 🟧 | >0.4 |
| 🟨 | >0.5 |
| 🟩 | >0.6 |
| 🟩 | >0.7 |

## DISCUSSION

### *Standard 1: The Context*

The goal of Standard 1 in this course is to situate the course content within the greater context of the software industry, and so survey questions related to the industry, and best practices of the profession (6, 7, 14, 15, 17, 18) were mapped to Standard 1. Average response scores to these questions were quite close to the overall average response score of 4.21, and thus we concluded the project, as well as the course content, were fairly well situated within the context of the conceive-design-implement-operate model, as well as the product-process-system lifecycle. One student commented that "developing a larger piece of software with multiple people that actually did something was very interesting and enlightening to do," indicating that they benefited as a learner from completing an industry-relevant project.

Students and professor comments from the additional questionnaire indicated the project was, however, somewhat lacking in the final phase (operate) of the CDIO context. One student noted, the "Project was relevant to some extent as it was trying to emulate a server-client architecture which is commonly used in web development. I wish we had more time learning how to authenticate and run the application on two sets of machines." The professor emphasized that, since the course was indeed a software *development* course, the focus was on industry-relevant skills including iterative development, requirements analysis, and design, the use of tools such as IDEs and Git and emphasized good design, software engineering best practices (such as SOLID principles), and the client-server architecture. However, the latter stages of the product-process-system lifecycle are not the focus: the project did not include a testing phase. Additionally, system requirements were simplified in order to make the project manageable.

### *Standard 2: Learning Outcomes*

Although not all CDIO learning outcomes were or even can be covered by a single software development course, the term project ensures many of them are (Crawley, Malmqvist, Lucas, & Brodeur, n.d.). Outcomes 1.1 and 1.2 are referred to in question 2, which was one of the lower-scoring questions. This indicates students did not feel as though they had quite enough knowledge going into the course as they might have liked to. However, questions 3, 5, 6, 7, and 8 all scored quite high, and these questions refer to outcomes 1.3, 2.1, and 2.2 since they cover a deeper level of knowledge and thinking that students acquire during the course.

The majority of the CDIO learning outcomes are covered through the term project (questions 13-18), in which they were encouraged to apply their disciplinary knowledge, problem-solving, system thinking, as well as work through at least the conceiving and designing aspects of the CDIO process. However, students were not only expected to code a fairly complex client-server application, but they also had to apply communication and presentation skills during demos and while communicating with their group members or instructors. As such, 'soft' skills were also required to successfully complete the project.

One student found that, with respect to group work, they had a fairly easy time with communication because they worked with friends. Another noted "when working in a group, it's good to have each person work on the individual classes/functions separately," indicating they gained some valuable insight on working with others. The professor commented on the differences he noticed between students who worked alone versus those who worked in groups, saying students who worked individually missed on the experience of teamwork and learning from their peers. The projects done in teams generally had more features, and were better designed." One drawback of teamwork, however, was "maintaining the balance of the work." Instructors worked to support students by discussing with groups and introducing best team practices - however, the distribution of group work is a very common problem, and it forces students to practice their interpersonal skills.

The main CDIO learning outcomes that were notably lacking in the course included 2.4 and 2.5 (attitudes, thought and learning, and ethics, equity, and other responsibilities) - though these could be argued to at least be touched upon for students who practiced best team practices - as well as 4.1 (external, societal, and environmental context), 4.5, and 4.6 (implementing and operating). These are most likely not covered in the course due to time and resource constraints.

### *Standard 5: Design-Implement Experiences and Standard 7: Integrated Learning Experiences*

The term project acts as both a design-implement experience and an integrated learning experience, as they are defined by CDIO. As such, the survey questions about the project are used to evaluate both Standard 5 and Standard 7 jointly.

Many of the working parts of a successful design-implement and integrated learning experience are outlined in Standard 2. For example, the simultaneous development of both disciplinary and interpersonal skills developed through the completion of the project are key attributes of an integrated learning experience. Further, the CDIO process (learning outcomes 4.3-4.6) is an essential aspect of an effective design-implement experience.

Overall, students appeared to benefit quite significantly from the term project. The survey included two short answer questions at the end, asking students what aspects of the course they found useful and how they would improve the course. Of the 43 responses, six students said the term project was useful, and fifteen said the same about labs (concepts and code from labs were later reused and built upon in the project). Three students commented that they wanted more group work or a more involved final project, while only one said that they didn't like the project. These comments, along with high response scores for the questions about the term project, are all indicative that students benefited significantly from the introduction of an integrated, design-implement experience. Additionally, although we recognize that many factors can impact student course ratings, the professor noted that the course saw improved ratings after the redesign.

### *Standard 8: Active Learning*

Active learning was an important and consistent theme across the duration of the course. Evidently, the term project is the aspect of the course that best reflects this theme, and was designed and implemented in order to employ the principle of active learning in a manner that would be manageable and set reasonable expectations for students. Students were presented with the challenge of applying the majority of the concepts and skills they had learned throughout the duration of the course but were not required to seek out *new* information in order to successfully complete the project. Two students mentioned having to do minor research on databases, but the third said they were able to complete the project without outside help, reaffirming the professor's assertion that the project was designed to be completed solely using the information from the course. The professor also provided a number of additional resources, including helpful links and instructional videos he created. Thus, while students were able to engage in active learning in the project, they were adequately supported and had all the resources they needed. Rather than the project being fully self-directed, students were guided into successfully completing it, thus keeping the scope of the project feasible for engineering students.

Beyond the project, lectures and labs were designed to encourage student participation. Questions 9 and 12 of the survey (instructor stimulated student interest and course organized to allow all students to participate fully) were among the highest-scoring questions.

## CONCLUSIONS

### *Implications*

Overall, we view the introduction and implementation of a term project as a success. Students benefited from the opportunity to develop a relatively industry-realistic client-server system by gaining (partial) experience in the conceive-design-implement-operate model. In addition, students engaged in the product-process-system lifecycle and worked towards many of the CDIO learning outcomes. Especially in the context of a single software development course, the term project successfully covered much of the relevant CDIO standards.

However, future iterations of the term project may benefit from a few changes. Firstly, because students who worked in groups developed their interpersonal skills far more than the students who chose to work individually, students should be encouraged, if not required, to work with others or at least complete part of the project in a team. Further, the project could benefit from being more precisely situated within the context of industry and especially the implement and operate stages of the CDIO context. This would provide students a more true-to-real-life design experience, as well as to adhere more closely to CDIO standards.

In terms of changes to the delivery of the course, some of the responses to the survey indicated that there were certain aspects of the project that were not taught at a sufficient level of detail, while others noted that lectures could seem disorganized or unfocused. In future iterations of the course, course content and lectures could be structured to focus on teaching specific aspects of the project (for example, the GUI, working with databases, or class relationships).

### *Limitations and future research*

The data used for this paper does have some limitations. Firstly, participation bias skews the results to be slightly more positive, and in particular, qualitative responses from the questionnaire are largely reflective of the opinions of strong students. Ideally, in the future we would be able to provide additional incentive to encourage more students to respond to the survey so would be able to hear more from students who are struggling, or who appear to be neutral to the structure of the course. Additionally, though the survey was anonymous, it would be beneficial for future iterations of the survey to be conducted by a neutral third party, so students would feel more comfortable responding openly.

Future research could include conducting an updated survey that more specifically targets CDIO standards and learning outcomes in order to monitor how student performance may change as the implementation of the project matures. This form of engineering education research, using relevant CDIO standards as a framework in order to analyze the effectiveness of different learning methods and projects, could be customized and applied to different engineering courses as well.

## REFERENCES

CEAB Graduate Attributes. (n.d.) Retrieved from https://engineerscanada.ca/sites/default/files/Graduate-Attributes.pdf

Cloutier, G., Hugo, R., & Sellens, R. (2012). Mapping the Relationship Between the CDIO Syllabus and the CEAB Graduate Attributes. *International Journal of Quality Assurance in Engineering and Technology Education.* https://doi.org/10.4018/ijqaete.2012040104

Cohen, L., Manion, L., Morrison, K., Cohen, L., Manion, L., & Morrison, K. (2018). Choosing a statistical test. *In Research Methods in Education.* https://doi.org/10.4324/9781315456539-44

Crawley, E. F., Malmqvist, J., Lucas, W. A., & Brodeur, D. R. (n.d.). *The CDIO Syllabus v2.0: An Updated Statement of Goals for Engineering Education.*

Crawley, E. F., Malmqvist, J., Östlund, S., Brodeur, D. R., & Edström, K. (2014). *Rethinking engineering education: The CDIO approach, second edition. In Rethinking Engineering Education: The CDIO Approach, Second Edition.* https://doi.org/10.1007/978-3-319-05561-9

Final Year Engineering Students 2017 Survey - National Results, n.d. Retrieved from https://engineerscanada.ca/reports/final-year-student-exit-report/2017-survey-national-results#the-graduating-class-of-

Grant, C., & Osanloo, A. (2014). Understanding, Selecting, And Integrating A Theoretical Framework In Dissertation Research: Creating The Blueprint For Your "House." *Administrative Issues Journal Education Practice and Research.* https://doi.org/10.5929/2014.4.2.9

May, E., & Strong, D. S. (2011). Is Engineering Education Delivering What Industry Requires. *Proceedings of the Canadian Engineering Education Association.* https://doi.org/10.24908/pceea.v0i0.3849

Mead, N. R. (2009). Software engineering education: How far we've come and how far we have to go. *Journal of Systems and Software.* https://doi.org/10.1016/j.jss.2008.12.038

Prime, Z., Robertson, W., Cazzolato, B., Missingham, D., & Kestell, C. (2015). Using The Honours Project Course To Enhance Engagement Across All Stakeholders. *Proceedings of the Canadian Engineering Education Association.* https://doi.org/10.24908/pceea.v0i0.5896

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education.* https://doi.org/10.1002/j.2168-9830.2004.tb00809.x

Pucher, R., & Lehner, M. (2011). Project Based Learning in Computer Science - A review of more than 500 projects. *Procedia - Social and Behavioral Sciences.* https://doi.org/10.1016/j.sbspro.2011.11.398

Puka, L. (2011). Kendall's Tau. *In International Encyclopedia of Statistical Science.* https://doi.org/10.1007/978-3-642-04898-2_324

Ríos, I. D. L., Cazorla, A., Díaz-Puente, J. M., & Yagüe, J. L. (2010). Project-based learning in engineering higher education: Two decades of teaching competences in real environments. *Procedia - Social and Behavioral Sciences.* https://doi.org/10.1016/j.sbspro.2010.03.202

Schramm, C., & Chan, A. D. C. (2013). Capstone Project Evaluation – Towards a Student-Centred Approach. *Proceedings of the Canadian Engineering Education Association*. https://doi.org/10.24908/pceea.v0i0.4840

Shaw, M. (2000). Software engineering education: A roadmap. *Proceedings of the Conference on the Future of Software Engineering, ICSE 2000*. https://doi.org/10.1145/336512.336592

Shekar, A. (2014). Project-based learning in engineering design education: Sharing best practices. ASEE Annual Conference and Exposition, Conference Proceedings.

Stoicoiu, C., & Cain, K. (2015). Industrial Projects in a Project-Based Learning Environment. *Proceedings of the Canadian Engineering Education Association.* https://doi.org/10.24908/pceea.v0i0.5903

Zouganeli, E., Tyssø, V., Feng, B., Arnesen, K., & Kapetanovic, N. (2014). Project-based learning in programming classes - The effect of open project scope on student motivation and learning outcome. *IFAC Proceedings Volumes (IFAC-PapersOnline).*

## BIOGRAPHICAL INFORMATION

**Dina Shoham** is an undergraduate student in Software Engineering at McGill University, who worked with Dr. Moshirpour and Robyn Paul as an educational research assistant.

**Robyn Paul** is a second-year Ph.D. student at the Schulich School of Engineering at the University of Calgary. Her work is looking at using best practices from ecofeminism to deconstruct the culture of engineering education and bring awareness to engineering's hidden curriculum. Robyn also has a master's degree in engineering education, where she studied engineering leadership education.

**Dr. Mohammad Moshirpour** is an instructor and Chair of Software Engineering Teaching Innovation at the Department of Electrical and Computer Engineering at the University of Calgary. His research interests include the areas of software architecture, software requirements engineering, data mining, and machine learning, and Software engineering, and computer science education. He is a senior member of IEEE and is the IEEE Chair of the Computer Chapter of the Southern Alberta Section.

### *Corresponding author*

Dina Shoham
McGill University
dina.shoham@gmail.com