Curriculum-Aligned Work-Integrated Learning: A New Kind of Industry-Academic Degree Partnership

Gail Carmichael
Shopify
Ottawa, Ontario
gail.carmichael@shopify.com

Christine Jordan
Shopify
Ottawa, Ontario
christine.jordan@shopify.com

Andrea Ross
Shopify
Ottawa, Ontario
andrea.ross@shopify.com

Alison Evans Adnani
Shopify
Ottawa, Ontario
alison.evansadnani@shopify.com

ABSTRACT

Work-integrated learning is a common approach to add practical, real-world work experience to academic settings. Traditional co-op programs in colleges and universities alternate courses with semesters spent as an intern at a relevant workplace. We have designed an academic-industry partnership that takes work-integrated learning further by deliberately aligning workplace experience to the academic curriculum. Our students earn a Bachelor of Computer Science from the university, and are paid employees of the industry partner throughout their degree. While advanced courses and electives are taken on campus as usual, some core computer science classes and practicum courses are delivered with the industry partner so as to integrate them with placements on site. Assessment remains the responsibility of the university. In this report, we describe the partnership from the perspective of the industry partner. We describe our goals, partnership design, and first two iterations of the implementation. We discuss the challenges we have faced with our first cohort, and share suggestions for others looking to create similar programs.

ACM Reference Format:
Gail Carmichael, Christine Jordan, Andrea Ross, and Alison Evans Adnani. 2018. Curriculum-Aligned Work-Integrated Learning: A New Kind of Industry-Academic Degree Partnership. In *Proceedings of The 49th ACM Technical Symposium on Computer Science Education, Baltimore, MD, USA, Feb. 21–24, 2018 (SIGCSE '18),* 6 pages.
https://doi.org/10.1145/3159450.3159543

1 INTRODUCTION

The educational needs of the twenty-first century's workforce are constantly in flux as scientific breakthroughs and new technologies redefine the nature of work. Educational institutions are not structured to be immediately responsive to changing skills demands, and require time to redesign and rejuvenate their curricula. Not enough students are graduating with the skills necessary to meet the technology industry's workforce needs [6]. The gap between skills provided by higher educational institutions and those that are needed in the workforce is especially apparent in computer science education, where programming languages, tools, and best practices are constantly evolving. The challenges facing computer science education relating to its rapid evolution along with the employability of graduates has been a topic of discussion and debate for many years [2, 5].

Several work-integrated learning models have been implemented in post-secondary educational institutions, such as industry-focused capstone projects and cooperative education projects. One way that work has been integrated with student learning is by having students perform capstone projects working with industry. A capstone project typically happens in the last year of a student's degree, as a culminating project that builds upon the knowledge the student has gained throughout the course of their degree program. An example of this is the Undergraduate Capstone Open Source Projects (UCOSP) [1] program that allows Canadian undergraduate computer science students to work on open source software projects for academic credit. With capstone projects happening at the end of a degree, it may be several years until students are able encounter the real-world applications of the theory they are learning. A disadvantage of this approach is that students do not get the motivational benefit of problem-based learning when a concept is introduced.

Cooperative education has been a popular model of work-integrated learning, but has remained mostly unchanged since its initial inception in 1957 [4]. It provides students with alternating learning terms at an academic institution and work terms in the workplace. Depending on the program, students may complete all their working terms at the same employer, or acquire a variety of work experiences by changing employers each term. It benefits students by allowing them to connect with their community of practice and work on real-world applications. However, there are several short-comings of the cooperative education model; in particular, the work performed in work terms is not guaranteed to reinforce the concepts that students have been learning or align with students' coursework. With learning and work terms being performed consecutively, students must wait months before being able to see the applications of the theory they have learned in class.

SIGCSE '18, Feb. 21–24, 2018, Baltimore, MD, USA

ACM ISBN 978-1-4503-5103-4/18/02.
https://doi.org/10.1145/3159450.3159543

Shopify and Carleton University have partnered to deliver Carleton's Bachelor of Computer Science degree in a whole new way. Instead of separating coursework and work placements, students in the program are immersed in the work environment and have the opportunity to connect the two deliberately and deeply. We call our approach curriculum-aligned work-integrated learning (CAWIL). In this report, we describe our partnership from the perspective of the industry partner. We introduce details of CAWIL in Section 2, including goals and the high level design of the partnership. In Section 3, we discuss the first and second iterations of implementing the partnership. We conclude in Section 4 with some suggestions for those considering their own CAWIL programs based on lessons we have learned so far.

2 CURRICULUM-ALIGNED WORK-INTEGRATED LEARNING

2.1 Partnership Goals

It is our philosophy that excellent software developers come from many types of backgrounds, and that a background incorporating both theory and practice is particularly valuable. We decided to partner with a local university to try a new approach to delivering a Bachelor of Computer Science degree. The university covers the theoretical aspects of the program as usual, while we not only provide relevant work experience, but deliberately align workplace competencies with the material learned on campus. We call our model **curriculum-aligned work-integrated learning** (CAWIL).

Our goal is to create a model that is different from other computing education initiatives. We are not trying to replicate the idea of a bootcamp, where the focus is much more heavily on specific industry technologies. We are also not offering an apprenticeship in the traditional sense, though there are aspects of apprenticeship in our team placements. We are closest to the idea of co-op. However, we are not alternating team placements with semesters on campus and hoping that students see how theory from courses can be applied on the job. Instead, students work and study at the same time, and we explicitly connect the two.

We consider a successful curriculum-aligned work-integrated learning experience to have the following characteristics:

- student coursework is deliberately and intentionally connected to team placements
- students have opportunities to master academic learning outcomes through their team placements and our in-house supporting curriculum
- whenever possible, students experience a concept in a practical setting through their team placements before they see the associated theory
- the time that passes between experiencing a concept and learning the theory is minimized
- students are given formal opportunities to reflect on connections between theory and practice

Shopify is a software company providing a commerce platform to entrepreneurs. Shopify's priorities naturally focus on its business. Broadly, our business goals are:

- increase the pipeline of **talented, qualified computer science graduates** to our company and more generally in Canada, all with a favorable opinion of Shopify as an employer
- improve Shopify's ability to **onboard and accelerate the impact of junior employees**, thus widening the pool of talent we can recruit
- achieve these goals in a way that delivers the **highest possible return on investment** (for example, by having our students contribute to the company in a meaningful way by the end of their degree)

To reach our business goals, we strive to support our students in becoming:

- **competent computer scientists** – Our students should build strong theoretical knowledge and deep experiential wisdom, and see meaningful connections between the two.
- **co-creators of an exceptional learning experience** – Students and educators should co-create a learning environment that is active, learner-centered, and problem-based.
- **resilient and in control** – Students should understand what is expected of them throughout their degree. They should not feel overwhelmed by the program's difficulty, and be able to maintain a reasonable work/life balance. They should feel well supported, for example through on-site coaching.
- **cultural role models** – Students should be able to fit well into the company's culture and be committed to its mission.
- **empowered and valued employees** – Students should see that they are contributing to the company through their work, and not feel indebted when they finish school.
- **connected collaborators** – Students should build camaraderie among their cohort, and become well connected both within and outside of the company.

In the following, the design of our partnership, and our experience implementing it over the past year, should be read in the context of these goals.

2.2 Partnership Design

Both Carleton University and Shopify contributed to the design of our partnership. On the university side, changes were required to accommodate the team placements in a way that supports CAWIL. Fortunately, we did not require the creation of a new degree program. Instead, we were able to add a special option to the existing Bachelor of Computer Science Honours program. It is very important to note that Carleton is still granting the degree, not Shopify. Carleton fully owns the assessment of students and assignment of grades. The same standards apply to all students, whether part of our partnership or not.

Students participating in the partnership are paid Shopify employees throughout their degree, both through a salary and tuition grant. There is no need to take on a part time job during school, and students can graduate debt free. After some time focused on learning fundamental computing skills, students join development teams and contribute to our products while learning competencies in a practical setting. After joining teams, students spend time both studying and working, ideally integrating practice and theory as per the goals of CAWIL. They are able to finish their degree in four years, and graduate with at least as many hours of work experience as a typical co-op program that usually takes five years to complete.

Most credit requirements in the degree remain the same, except that eight free electives become practicum credits. Practicums are pass/fail courses that capture the industrial skills students are learning at the company, both through their team placements and through educational material developed at Shopify. Practicums are also used to give credit for the acquisition of soft skills and exploration of other CS topics of interest. Two practicums are covered each year, and each year has a theme.

The degree includes eight core computer science courses that all students take, including CS1, CS2, and web development fundamentals. It is through the core courses that we want to tie theory and practice together especially tightly. We aim to have the competencies from these courses mastered on site at Shopify through active learning workshops, self-directed learning, and work on team placements.

Math courses, advanced computer science electives, and breadth electives are taken on campus as usual. We encourage students to take breadth electives later in their degree than is typical, pushing CS content earlier. We do this to ensure students are ready to join teams as soon as possible, and to help students find electives later on that can connect to their computing skills and interests.

Students interested in participating in our partnership for their degree must apply for one of a limited number of places. Students apply first to the university, and if they are accepted to the Bachelor of Computer Science Honours program, they submit an application portfolio to Shopify. We screen students based on their applications, then interview them in a process quite similar to other Shopify interns. We look for students who have a demonstrated passion for wanting to solve real-world problems with computing, along with a strong academic background and/or at least some prior experience with computing. We believe strongly in the value of diversity, and work hard to encourage underrepresented groups in tech to apply.

Accepted students can expect to receive extra support from Shopify staff. Our team provides coaching before students join teams and instructional support for material delivered on site. We help them get used to what is quite a new way of learning for most of them, and help them navigate a career as a software developer – this will be the first job many of our students have had. Placement teams provide mentors and coaching after students start working.

3 PARTNERSHIP IMPLEMENTATION

Shopify welcomed our first cohort of students in September 2016. They experienced the first implementation of the partnership design described above, and encountered our first set of challenges. Both are discussed below as the alpha version of the implementation. We took what we learned from our first cohort and designed the beta version of the partnership, which will partially be implemented for the first cohort and fully implemented for the second cohort that joined us in September 2017.

3.1 Alpha Implementation

Our first cohort of students joined us in September 2016 for the alpha implementation of our program. We had eleven students total (six female, five male). Four were mature students, returning to school having completed another degree some time in the past. We tested the alpha implementation design for two semesters before transitioning to the beta design.

Figure 1 summarizes the structure of the alpha implementation. Students took up to three courses and a practicum each semester, including over the summer (on-campus students typically take five courses each in the fall and winter semesters). Team placements began in the second semester, when material for the core computer science courses was learned through placements as well as workshops and self-paced learning delivered on site. Students would be working on teams every semester after that except for two, when they would attend courses on campus full time.

One of our most important goals is to increase the number of talented software developers primarily at Shopify, as well as in the greater community in Canada. As such, we designed an unconventional path through Carleton's Bachelor of Computer Science. Students typically take CS1 and CS2 in their first and second semesters respectively, which then opens up a number of core courses that can be taken in second year; math courses and breadth electives fill out most of the first year. We wanted to accelerate when our students acquired fundamental computing skills so they could meaningfully contribute to the company as soon as possible. Delivering both CS1 and CS2 back to back in the first semester helped meet this goal, as did getting students onto placement teams as fast as possible.

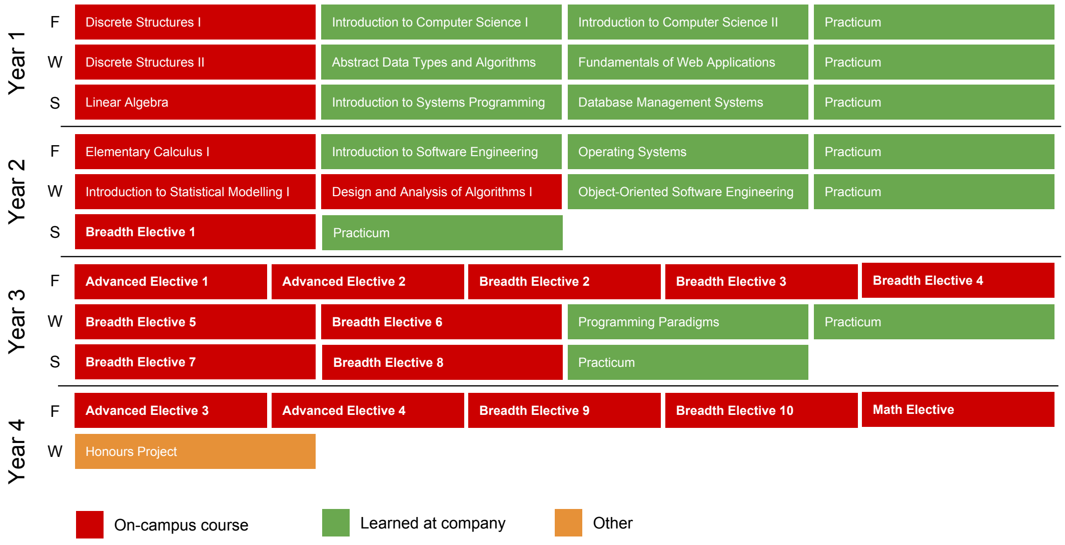
Not shown in Figure 1 is a Ruby on Rails bootcamp given to students at the beginning of the second semester. After the two week long intensive introduction to the technology used on web-centric teams at Shopify, students started their first placements. In the second semester, web development and data structures competencies were meant to be mastered through work on teams, self study, and active-learning workshops delivered on site. Students participated in the assessments given on campus. The plan was for students to rotate their team placements every two semesters with the goal of performing work closely related to the competencies of the core courses delivered in that timeframe.

Mentoring was structured such that each student received one-on-one time with a mentor from the education team. During biweekly meetings (and additionally as needed), the mentor and student discussed academic progress and personal development. The student was also encouraged to provide feedback about the partnership. Mentors also met with each other biweekly to discuss the topics to cover in the next sessions and to raise any issues that need to be addressed by the team.

3.2 Alpha Challenges

As is to be expected with an undertaking this large, we ran into several challenges with the alpha implementation of our partnership. Some of these were due to time and staffing constraints on the company side, and others indicated where we could improve our design. We collected data from our mentorship and instructional sessions, interactions with education staff, and formal anonymous surveys sent regularly to students to measure against the goals of Section 2.1.

Our first semester with CS1 and CS2 was largely successful. Students performed well in the online, self-paced courses and generally maintained a reasonable balance with their on-campus course, their



Year 1	F	Discrete Structures I	Introduction to Computer Science I	Introduction to Computer Science II	Practicum	
	W	Discrete Structures II	Abstract Data Types and Algorithms	Fundamentals of Web Applications	Practicum	
	S	Linear Algebra	Introduction to Systems Programming	Database Management Systems	Practicum	
Year 2	F	Elementary Calculus I	Introduction to Software Engineering	Operating Systems	Practicum	
	W	Introduction to Statistical Modelling I	Design and Analysis of Algorithms I	Object-Oriented Software Engineering	Practicum	
	S	**Breadth Elective 1**	Practicum			
Year 3	F	**Advanced Elective 1**	**Advanced Elective 2**	**Breadth Elective 2**	**Breadth Elective 3**	**Breadth Elective 4**
	W	**Breadth Elective 5**	**Breadth Elective 6**	Programming Paradigms	Practicum	
	S	**Breadth Elective 7**	**Breadth Elective 8**	Practicum		
Year 4	F	**Advanced Elective 3**	**Advanced Elective 4**	**Breadth Elective 9**	**Breadth Elective 10**	**Math Elective**
	W	Honours Project				

🟥 On-campus course 🟩 Learned at company 🟧 Other

Figure 1: Course sequence for alpha implementation.

practicum, and getting settled into life at Shopify. We had originally planned to deliver our Ruby on Rails bootcamp for three to four weeks at the end of the semester, but were unable to dedicate time to its development. Instead, we held a two week bootcamp created by another Shopify employee at the beginning of the second semester.

Students joined teams immediately after the bootcamp, but were not sufficiently prepared. The bootcamp was too short, and could have been better designed pedagogically. Further, students were not proficient enough in GitHub, which we intended for them to use for their practicum projects and which was covered again only superficially during the bootcamp. In some cases, students had not developed strong enough programming skills for the particular teams they were joining.

Teams were not sufficiently prepared to accept our students, either. It took us an unexpectedly long time to find teams willing and able to take on our relatively junior students when they were used to interns further on in their education. As a result we ran out of time to prepare the teams that did agree. We did not set proper expectations of what level our students would be at and what kind of work they should do. Developing scaffolds to be used in the future would benefit not only our own students, but the ability of all teams across the company to take on more junior employees.

Delivery of our two second semester courses – web development and data structures – failed to meet our goals of CAWIL. Students had the option of participating in just the final exam for each course, but did not want their entire mark based on one test. They opted to take the midterm as well as submit assignments, the latter to help prepare for the former. Unfortunately, by participating in all the assessments, we were very tightly tied to the cadence of the on-campus course, yet we did not get information from instructors any sooner than the students. In the case of the web development course, we also found ourselves tied to the technologies used on campus, which did not match what we used in house. Students struggled with the self-directed learning we assigned them and yearned for more structure. Because of these challenges, placement work did not contribute directly to learning the core CS courses and we were unable to spend time in workshops making connections to it.

The difficulties of the second semester negatively affected our students' well being. Joining teams too early caused them a large amount of stress and anxiety. We discovered that their trust in the education team diminished. With team members acting as mentors as well as instructors, students were not forthright and often did not bring forward their actual concerns, or tell us if they were in crisis. Additionally, most students coming from high school were experiencing their first professional job experience. Not all knew how to behave in some situations without additional coaching, sometimes escalating situations further.

One interesting issue that arose that may have contributed to some of the other problems is that of student perception. After team placements began, we found that some students felt that they were getting "a raw deal" with this program. They compared themselves with full time engineering staff and felt badly that they were the lowest paid, with the least vacation and the most work on their plate. A group coaching session helped these students shift their perspective to that of their peers, who felt very lucky to be part of a program with many perks compared to other students taking the traditional path through the degree.

Finally, we found that our model for mentoring needed adjustment. As mentioned above, having our staff play the role of mentor, instructor, and program developer led to not receiving accurate information and feedback from students. The model would also not be scalable as we welcome new cohorts each year and expand with new partnerships.

3.3 Beta Implementation

After two semesters of the alpha implementation, we began working on the beta implementation design (simply referred to as beta

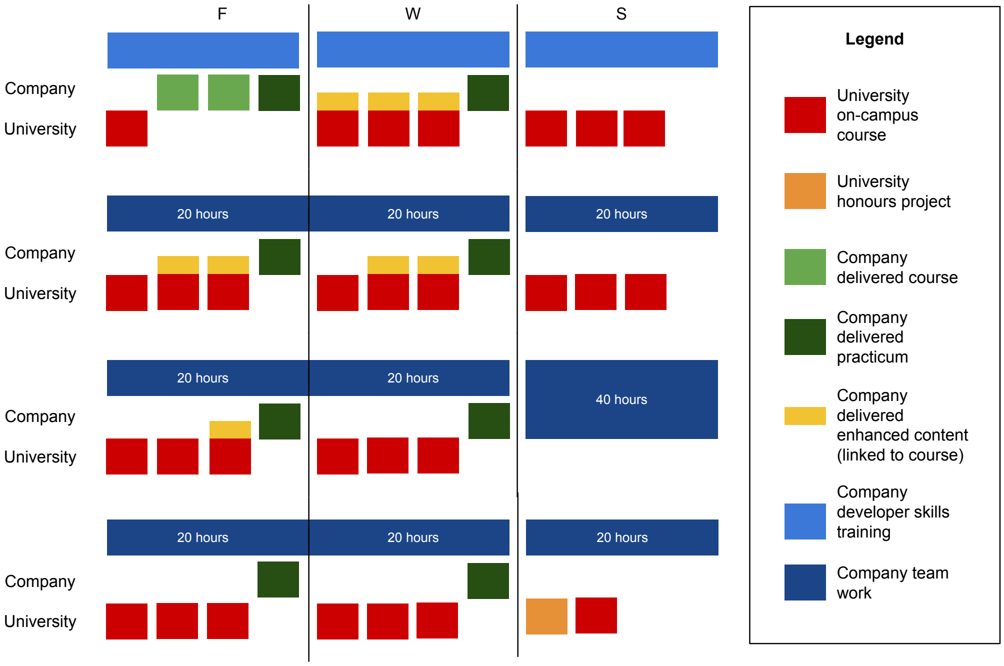


Figure 2: Course structure for beta implementation.

from here on) to address some of the challenges outlined above. Our second cohort of students began in September 2017 and will experience beta fully. Our first cohort of students will experience a modified version of beta that fits with what they have completed in alpha so far. The key areas of improvement for beta are centered around curriculum, placements, and student wellbeing.

With respect to curriculum, the most important changes bring us closer to meeting our CAWIL goals. The structure of beta is summarized in Figure 2. In beta, we focus our time on our greatest strengths: industry-specific knowledge and how it fits with theory. Rather than find ourselves watching what instructors are doing on campus and trying to replicate it, we will add a new layer between academic courses and work at Shopify. Starting second semester, students will take most core computer science classes on campus, and then engage with additional material and workshops at Shopify that link the theory they learn to industry practice. Practicums in beta capture this enhanced content layer as well as team placements.

CS1 and CS2 will remain as online courses run back to back first semester. However, the first practicum in beta works differently than in alpha. Instead of group projects of the students' choosing, students participate in developer skills training courses in topics such as command-line interfaces, and GitHub. Developer skills training continues in the second and third semesters, sometimes tied directly to on-campus course and sometimes as standalone topics.

Students will now begin their first team placements at the beginning of their second year. By pushing back when students join teams, we are better able to prepare them in terms of technical and professional skills. In turn, development teams will benefit from more productive students, and place higher trust in the value of the partnership. Students should also feel significantly less pressure to perform at too high a level early. Another benefit of later placements is the ability to better match students to the *right* team in terms of skills, personality, need for mentoring, and so on.

Perhaps most importantly, we have put a much greater focus on student wellbeing for beta. We saw first hand in alpha how important a positive student experience is, and that performance in both academics and work placement is significantly affected if student wellbeing degrades. As a result, we have created a dedicated role in our education team to facilitate a positive student experience. We have also designed a *Life at Shopify* program that matches students with mentors outside of the education team. The lead of the education team is additionally available to students, as is the lead of their work placements when those begin.

3.4 Data Collection

We currently have several mechanisms for collecting data about our program. In the future, we hope to conduct a more in-depth analysis of our data. At present, our data informs us on how students are performing and what changes we can make on an ongoing, realtime basis.

Student weekly reports: Students submit a weekly report on Fridays using a stoplight rating for wellbeing, academics, and work placement. Green signifies no significant issues, yellow signifies some issues but nothing serious, and red signifies serious issues. Students reflect on their work from the past week and share plans for the next week, what they need help with, and any other general comments.

Mentor bi-weekly reports: Each placement mentor submits a performance evaluation report every second Friday, again using a stoplight rating for wellbeing, academics, and work placement results. Mentors also numerically rate students on various dimensions ranging from dependability to impact.

Practicum term reports: Students submit summative practicum reports that tie academic learning objectives to work placements in the fall and winter terms and allow for reflection on work placements and the program as a whole.

Student life one-on-ones: About once a month, students have an informal meeting with one of our staff. Additional meetings may also occur if there are discrepancies in the above reports.

Check-in surveys: Quarterly surveys measure student well being and other characteristics that can be compared to other Shopify employees.

3.5 Future Directions

To fully realize our goals of CAWIL, we would like to fully take on delivery of the core computer science courses at Shopify. By doing so, time already spent on developer skills training and in work placements could go toward mastering the competencies of a particular course, and we would be able to use specific technology to illustrate the theoretical competencies. With a more flexible schedule, theory could be mastered at just the right time – ideally after gaining some practical experience to motivate the theory – through independent learning or workshops that could be developed in collaboration with the university.

Developing competencies is one way to achieve full adoption of courses while ensuring assessment remains with the university. We define a competency as skills or knowledge in a topic that through various learning experiences help a student achieve a defined learning outcome. Upon achieving the learning outcome, the student can claim a new level of mastery such as those inspired by ACM and IEEE's Computer Science Curricula [3]: familiarity, usage, and assessment. For each course to be delivered at Shopify, both partners would work together to list the competencies that all students should be mastering. When these are agreed upon, exams can be designed to test to these outcomes in a technology agnostic way. Then, how a student comes to master the competencies – whether on campus or at Shopify – no longer matters, so long as they can succeed on the exam.

As we develop the enhanced content layer for core courses, we hope to also start building a proposed set of competencies. Eventually, we hope to be able to move toward fully delivering the core courses on site.

4 CONCLUSION

We have described a new kind of partnership between academia and industry following what we call curriculum-aligned work-integrated learning (CAWIL). This special option to the Carleton's Bachelor of Computer Science program is delivered jointly on campus and at Shopify. Students are paid, have their tuition covered, and graduate with many hours of practical work experience that is tied to the theory learned in class.

After iterating on the program's design, we have several lessons learned that may help others considering a similar endeavor. Our suggestions are listed below, organized loosely by topic.

- General:
  - Consider experimenting with CAWIL for a semester with select core courses before trying an entire degree program.
  - Expect to hire a substantial number of skilled staff for your education team, and expect this not to be easy.
  - Recognize there are three pillars for a program like this (curriculum, work placements, and wellbeing). Pay close attention to wellbeing because if it degrades, the other two will follow.
- Curriculum:
  - Allow sufficient time to develop quality curriculum.
  - Look for ways to allow core computer science courses to overlap as much as possible with work placements (in the best case, meeting course learning objectives by delivering the learning experience completely at the industry partner).
- Work placements:
  - Carefully consider each development team's capacity for working with students and their availability for providing support through mentorship, supervision, checking of work, and so on.
  - Carefully consider the minimum set of skills and experiences required to join development teams and be successful. These will vary from team to team.
  - Formally and prominently link performance evaluation and prestige of those working with students to the successful development of the student. In other words, ensure student mentors get recognized.
- Student experience:
  - Students will tend toward self-reporting results that are more positive than reality. Mentors often struggle with giving complete and candid feedback. Students often struggle with receiving constructive feedback, causing a vicious cycle.
  - Consider proactively giving both mentors and students workshops on crucial skills such as giving and receiving feedback, forthright discussion, and other related communication skills.
  - Ensure students have mentors outside of the education staff and ensure they can speak freely through appropriate venues.

REFERENCES

[1] Undergraduate Capstone Open Source Projects. http://ucosp.ca/ [accessed August 2017].

[2] L Harris. No end to ICT skills crunch. *ITWeb Brainstorm Magazine*, pages 78–83, 2011.

[3] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA, 2013. 999133.

[4] Bruce A McCallum and James C Wilson. They said it wouldn't work: A history of cooperative education in Canada. *Journal of Cooperative Education*, 24(2-3):61–67, 1988.

[5] Yma Pinto. A strategy, implementation and results of a flexible competency based curriculum. *ACM Inroads*, 1(2):54–61, 2010.

[6] Alex Radermacher and Gursimran Walia. Gaps between industry expectations and the abilities of graduates. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 525–530. ACM, 2013.