



Computing Capstone Courses as Preparation for Practice: A Global Survey of Instructors

BY JACQUELINE WHALLEY, AUCKLAND UNIVERSITY OF TECHNOLOGY
ASANTHIKA IMBULPITIYA, OTAGO POLYTECHNIC AUCKLAND INTERNATIONAL CAMPUS
TONY CLEAR, AUCKLAND UNIVERSITY OF TECHNOLOGY



IMAGE MONTAGE, © WWW.ISTOCKPHOTO.COM/ANTON VIERETIN

Many institutions, like ours (a large metropolitan university of technology), run some form of undergraduate capstone computing course during the final year of study. As computing educators, we have all read research papers on running and designing such courses. These papers often present an ideal model for an ideal world. However, many of us face large cohorts, an industry or research context reluctant or unable to provide suitable capstone opportunities, and a general lack of resources. So, what do our courses really look like and what challenges are encountered in running a capstone project course?

In trying to answer these questions, we discuss the results of an exploratory global survey of computing academics involved in the delivery of capstone projects which maps their experiences with, and the landscape of, contemporary computing capstone courses in practice. Key characteristics of courses are explored, including whether their capstone courses are a mandatory component of their degree, typical team size and formation, course duration, degree of industry involvement, student support mechanisms, and assessment approaches. Survey participants related several core challenges faced when delivering capstones, including student preparedness, the mismatch between industry and academic expectations, difficulty sourcing projects, and industry contribution and buy-in. These challenges influence the way in which capstones are delivered.

Introduction

This article presents the results of an exploratory global survey of instructors, responsible for the design and delivery of computing capstone projects, and their experiences with capstone courses, which have been defined thus.

“Most computing curricula have a special course, usually taken during part of the final year, that is considered a “capstone” course. This course is required of all students and is supposed to provide a culminating and integrative educational experience.” [17]

In addition to the classic definition above, several variants of capstone courses exist—work-integrated learning (WiL) [16,42], internships [35], sandwich programs [39], co-operative education [16], and so on. Capstone courses have been a long-established part of computing pedagogy and curricula, with research extending back some decades [10,17,20,24,32]. However, determining the scope of a capstone project or a course can be challenging. If we consider the course as part of a process of developing a graduate prepared for professional practice, then it can be seen as one element of a greater work-integrated learning strategy. That ‘practice orientation,’ while not precluding ‘research as practice,’ has colored our findings in this study and reflects the emphasis in our questionnaire

in Appendix A on a capstone course as offering a form of “authentic learning” [30]. But we acknowledge that the project and team-based form of learning typical to ‘software development’ capstones may differ from other models of capstone with a different focus, and perhaps scale, toward integrating the curriculum. Individual research-based capstones, capstones linking with research teams, community-based capstones, computing-for-the-social-good studies [27], and critical and collaborative service-learning models [22], as an educational contribution to the student’s learning, may not have been fully captured in our study.

Our conception of a capstone course is compatible with that of Tappert and colleagues at PACE University [45], who have argued that *“The aim of our capstone course in computing is to familiarize students with how their trade is plied in organizations, so that the program of study delivers ‘the practice’ part of the promised ‘theory and practice.’”* Yet this approach to developing professionalism does not ignore the theoretical aspects of the computing discipline and profession. As observed in Frezza et al. [25], reflecting on the need for developing critical dispositions in students such as *“Professionalism/Work Ethic”*, the capstone course inherently aims to educate students to act professionally as *“trained and skilled people: Acting honestly, with integrity, commitment, determination and dedication to what is required to achieve a task”* [25]. So, the ethical issues, for instance, that may arise in the context of the project must be understood and dealt with in their context. To further place this in context, the elements that make up WiL have been depicted in a co-operative education continuum by Clear et al. [16].

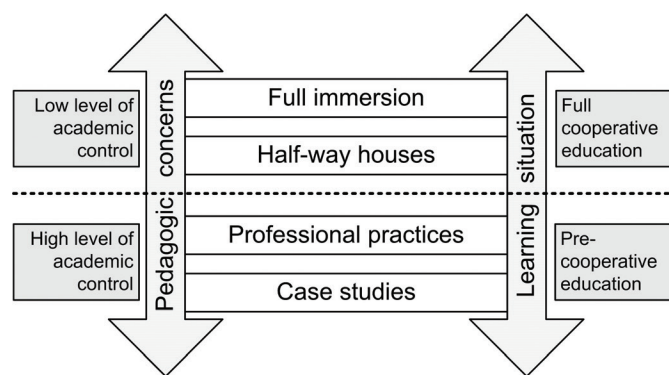


Figure 1: The cooperative education continuum – ex. [16]

The classic capstone course could be viewed as a “half-way house” in that model. “Full immersion” reflects onsite co-operative learning courses, sandwich programs, or internships. “Professional practices” and “Case studies” reflect pedagogical strategies which simulate professional work practices and contexts. The co-operative education continuum model in Figure 1 suggests that several barriers and

challenges exist for retaining academic control over pedagogic quality and consistency in the learning situation when running capstones. For instance, Wright has argued “locating real programming projects in the community...is often too time consuming. Furthermore, real projects are usually too difficult for college seniors...or students working on imaginary projects...may give students a poor understanding of the real world.” [49] Nonetheless, capstone courses are generally viewed as critical aspects of a computing curriculum. Professional societies and accreditation models aiming to set professionally informed standards for quality curricula exist in many countries, for instance Accreditation Board for Engineering and Technology (ABET) in the U.S., Information Technology Professionals New Zealand (ITPNZ) in New Zealand, Australian Computer Society (ACS) in Australia, and British Computer Society (BCS) in the U.K. In these curricula and accreditation models, a capstone course is a mandatory component. In this study, we are interested in the “half-way house” model, where projects have an industry or practice setting link but also have a level of academic control, support, and assessment that goes beyond that which is typically provided for “full immersion” models like internships, where the student works essentially in the role of an employee in a workplace with limited academic oversight. Our survey was designed with the intent of mapping the real landscape of experiences of instructors of contemporary computing capstone courses. We were interested in exploring what types and characteristics of the capstone courses our respondents deliver in practice and what challenges they encountered when running these courses.

Related Work

There is a wealth of literature on computing and software engineering capstone projects. In the last 10 years, most papers have focused on: curriculum [21,33]; use of agile/scrum [19]; project selection [6]; and student factors such as attitudes [38], values [48], and experiences [5]. Much of this research has been conducted in the context of a single institution. Our focus is on course characteristics and challenges in delivering capstones from a multi-institutional and course instructors’ perspective. The intent here is not to provide a comprehensive systematic review of the literature but to highlight key articles within these points of focus.

COMPUTING CAPSTONE COURSE CHARACTERISTICS

In 2015, Tappert et al. [48] surveyed 34 American universities about their capstone courses. In addition, they collated information on a further set of institutions’ capstone offerings from publicly available information on the Internet. The key information on the 49 institutions gathered resulted in a set of capstone course characteristics.

There is a wealth of literature on computing and software-engineering capstone projects. In the last 10 years, most papers have focused on: curriculum ... [and] been conducted in the context of a single institution. Our focus is on course characteristics and challenges in delivering capstones from a multi-institutional and course instructors’ perspective. The intent is not to provide a comprehensive systematic review of the literature but to highlight key articles within these points of focus.

We have collated these characteristics in Table 1 as a preliminary taxonomy for describing capstone courses.

Tappert et al. reported that most capstone projects were conceived by the clients: “33 of the 48 universities where this information was found—were generated by the project customers.” [45] In 10 of 45 universities (22%), “projects were determined jointly by the real-world project customers and the instructor, and in most cases (68%) involving external customers, the instructor had to approve the projects.” [45] Three modes for student team selection were reported—teams were either formed by the students or the instructor, or students were randomly allocated to a team. In cases where the instructor allocated students to teams, factors such as student work history, student expressions of interest, and geographical location were considered. Two assessment strategies emerged through their survey—regular progress reports and summative peer reviews. Surprisingly, nearly a third of the academics polled ran courses that did not require progress reports.

Table 1: Project dimensions from Tappert et al.’s survey. [45]

Dimension	Modes
Project Source	Industry Instructor Student
Student Team Selection	Self-Selected Randomly Selected Instructor Selected
Team Dispersal	In-person Dispersed-Hybrid Dispersed-Online
Assessment Strategies	Regular progress reports Peer Review

A 2023 systematic literature review (SLR) of software engineering capstone courses reported on the characteristics of “real-world” capstone projects through an analysis of 127

articles [46]. The result was a taxonomy of capstone project characteristics (see Table 2). Of the four dimensions discussed by Tappert et al. [45], three—project source, team selection approach, and assessment strategies—were also identified in this SLR. Interestingly, while Tappert [45] and Dugan [20] highlighted the importance of ‘student team selection,’ it was an attribute absent from the Tenhunen et al., [46] taxonomy of Table 2.

Table 2: A taxonomy of capstone project characteristics collated and adapted from Tenhunen et al.’s SLR. [46]

Project Attribute	Values
Course Duration	< 1 semester 1 semester 2 semesters > 2 semesters
Team Size	range: 1- 35 students norm: 4-5 students
Client	Internal External
Project Source	External Stakeholder† Instructor* Student
Technology Selection	Common technologies across projects Team selects dependent on project
Assessment Strategies	Peer evaluation Self-evaluation External client evaluation Course staff
Support Strategies	Instructor More experienced students Industry advisors (not client)

* Referred to in source [46] as Course Staff

† Note that the definition of external stakeholder in Tenhunen et al. [46] differs from that of an industry client in Tappert et al. [45]. Here external means external to the capstone course instructors, not necessarily to the institution.

Most of the courses reported in the SLR [46] ran over a single semester (66%), with the next most-common duration being two semesters (24%). These projects were typically conducted in teams of four to five students. According to Tenhunen et al. [46], a team of four to five students is the “sweet spot, cancelling out the negatives of the two extremes.” By a small margin, more projects (58%) were classified as being “external” with clients or product owners outside of the instructors on the course. Internal projects were defined as projects where either the client or product owner role was enacted by a course instructor or where students worked independently with some academic oversight of progress. Projects were typically sourced from external stakeholders (62%) and course staff (21%). Technology selection tended to depend on whether the teams were undertaking a common project. If the project was the same for all teams, then the technology tended to be specified by the instructor. In terms of

assessment, the most common end of project assessment involved instructor grading of project artifacts which provide evidence of both product and process. There were surprisingly few papers found that discussed the use of peer- and/or self-reviews. Most of the papers reviewed mentioned the importance of continuous support in the form of mentoring and coaching. Many courses used academic staff and course instructors to mentor teams. A few noted using industry advisors or more experienced students to guide teams.

Computing Capstone Course Challenges

Dugan [20] conducted a comprehensive SLR of the literature on software-engineering capstone courses in 2011. One area of focus was course and project issues. The challenges noted in the SLR included: time instructor needed to manage students and industry clients; industry projects being too large for one semester; risks associated with changes in client priorities; and the unique experiences student teams have when each is doing a different project. Fair assessment and assessing the contribution of individual team members is also recognized as a challenge for educators [37,47]. These findings offer a point of comparison to our own survey results, further discussed under the results section. Before discussing the results, the method adopted in survey design, conduct and analysis is now introduced.

Method

Participants were instructors involved in the design and teaching of final-year undergraduate capstone courses in computer science. We recruited participants through the SIGCSE mailing list. Overall, 54 participants started the survey. Of these, 18 were not included in the study because their institution did not offer work-integrated learning programs. The remaining 36 participants were from institutions in Australasia, Europe (including U.K. and Ireland), North America, South America, and West Africa. No two respondents were from the same institution (see Table 3). This study was approved by our institutional ethics board [22:203]. Participants provided written informed consent to participate in this study.

Table 3: Geographical regions of the survey participants.

Country	% Participants
Australasia	22
Europe	7
North America	63
South America	4
West Africa	4

SURVEY

The mixed-design survey (see Appendix A) included open-ended and closed questions. Questions comprised a list of potential categories, allowed multiple selections, and always included an "other" option to capture additional, unanticipated data. The survey was peer-reviewed and refined in collaboration with four local capstone course instructors. No demographic information was collected, and dichotomous (yes or no) questions were used to ensure subsequent questions were relevant to the experiences of the participant. The participant response question flow is illustrated in Figure 2. The survey was delivered online through the Qualtrics system for a period of two months. Geo-locations were collected by default and Qualtrics was set to flag duplicate submissions.

ANALYSIS

The closed and list selection questions were analyzed using simple descriptive statistics. Our analysis of barriers and challenges (Q4) used inductive ('grounded in' the data) reflexive (interpretivist) thematic analysis [7-9,29]. In this form of analysis, it is important that homogeneity within the data is not assumed, and descriptions and illustrative examples be used instead of inter-rater reliability [9]. This method was chosen to best analyze the brief segments of data available, from a small number of participants, and draw out the origins of the barriers, and to challenge our assumptions in interpreting the data. The first author conducted the initial coding and theme development, and the second author contributed to the review, refinement, and naming of themes. The following iterative process of five steps was followed [7,34]:

1. Data familiarization
2. Generating codes
3. Generating themes
4. Reviewing themes
5. Writing up with analytic narrative and extracts from survey responses

When analyzing the brief open responses to sub-questions on the key features of the respondents' capstone courses (Q6), the text was read to identify relevant

information, which was then extracted. Synonymous terms were merged, and the extracts labeled. For example, in assessment practices (Q6), mentions of practices or processes that involved peer assessment, peer review, or peer feedback were manually tagged as "Peer Feedback" and then quantified.

Reflexive thematic analysis [9] is a process that is seen through the lens of the researchers and involves a reduction of participants to their written responses and our interpretations of those textual expressions. Thus, knowledge is created through interaction with the participants and the data, so it is important to acknowledge the perspectives and positioning of the researchers who performed the thematic analysis for interpretation of the findings. We have both taught and led our institution's capstone course. We mentor several teams each year. Our projects are year-long external industry projects, sourced by the academic team and vetted for suitability. The teaching team, along with our industry liaison administrator, collaborates with industry partners to frame projects that are appropriate for our capstone. We do not offer other forms of WiL in our computer- and information-science bachelor's degree. Teams are closely mentored by academics and have regular contact with their clients. In some cases, teams work at their client's site. We have one full day a week in the student's schedule blocked out in their timetables and weekly workshops focusing on, among other topics, project management, risk management, communication, and collaboration.

Results

The survey of academics was framed through the following invitation: *We are interested in exploring your experiences with tertiary programs/courses designed to develop work-ready graduates. We are interested in the full range of models designed to help students transition into their first professional role from one-on-one mentorship to capstone projects and work-based-learning/internship programs. We would like to invite you to participate in a study in which we explore any initiatives that you and/or your institution are involved with that help to onboard these new graduates or work-ready students.*



Figure 2: Participant question response flow. See Appendix A for questions (Q1-6) in full. Number is the count of responses at each question node. Work-integrated learning is abbreviated to WiL.

The first question in our survey asked which models of authentic learning the participants use to prepare their final-year undergraduates for professional roles in industry. As noted in the introduction, this ‘professional practice’ focus, while accepting ‘research as practice’ as an option, has quite possibly served to limit the scope of the responses and findings to more software development and team-based capstone models.

LEARNING MODELS

Having made that observation, the three most common models of learning for the initial 54 respondents were project-based learning, case studies, and role play (see Figure 3). Recall that more than one option may be selected by a respondent, so the number of models identified exceeded the number of responses (N).

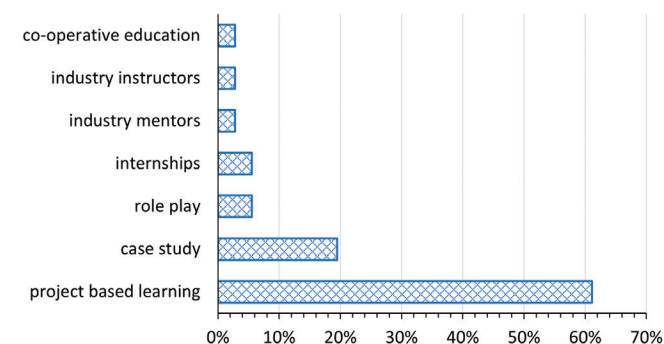


Figure 3: Models of learning (Q1) (N = 54).

In all cases the capstone projects formed a mandatory component of the degree. Two respondents noted that their capstones were only mandatory for the students majoring in software engineering. Our Australian respondents mentioned that for them capstones are a core component of their bachelor degree program’s ACS accreditation.

Other models of learning included engaging working industry professionals as teachers and industry mentorship programs. Just under half of the respondents who used project-based learning also used other approaches, the most common being case studies and/or role play. Figure 4 shows the combinations of authentic models of learning respondents reported. The left horizontal bar chart shows the number of responses that included each of the models of learning (attributes) listed. For example, five responses included internships so five is the set size. Following this row, the right-hand matrix illustrates the intersections (highlighted with a black box in Figure 4). The top vertical bar chart shows the number of responses for each of the intersections. For the internships, there are two intersections: internships alone and internships in combination with project-based learning and case studies.

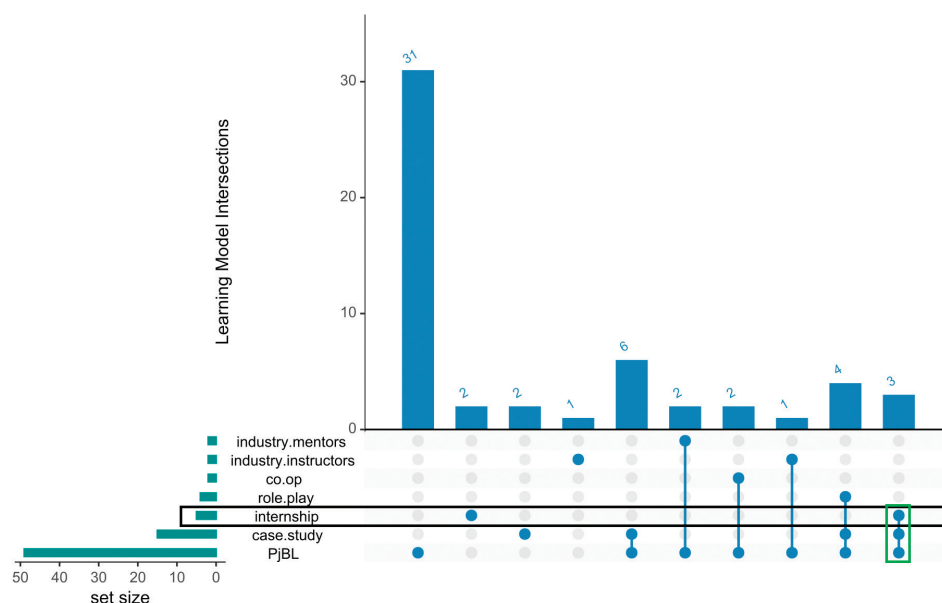


Figure 4: UpSet diagram, created using UpSetR [18], of the sets of authentic learning models used. Black horizontal box illustrates the internship attribute and its set size. The vertical blue box highlights the intersection of the attributes internship, case study, and project-based learning, which has a learning model intersection of three responses as indicated in the top bar chart.

TYPES OF WORK-INTEGRATED LEARNING PROGRAMS

Of the initial 54 respondents, 67% (36) came from institutions that offered WiL programs for their undergraduate students (Q2). Only 11 (31%) of these respondents indicated that their institute offered industry-linked capstone projects (Figure 5). Given that many accreditation programs (for example, ITPNZ, ACS [1]) require a capstone project with a preference for a team project conducted in conjunction with a real industry client, the low occurrence of such courses was unexpected. The most common model offered was that of unpaid internships/work experience. Four courses included virtual or simulated work experience as a component of their program, which included either industry-linked capstones or work placements. None of the institutions offered sandwich programs.

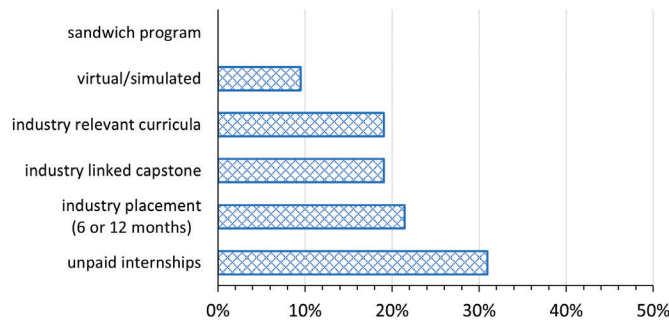


Figure 5: Programs offered (Q3) (N = 36).

CAPSTONE COURSE/PROJECT OFFERED

Overall, 83% (45) of respondents offered capstone projects for their final-year undergraduate students. Two of the 36 participants who reported they offered WiL (Q2) did not offer final-year capstone projects (Q5) but instead ran work-experience programs in the form of unpaid internships. The remaining 34 participants offered WiL (as per Q2) and final-year undergraduate capstones (Q5). Eleven of the respondents who did not offer WiL or industry-based capstone projects (Q3) reported that they did offer capstone projects (Q5). Twenty-six of the 45 respondents who offered final-year undergraduate capstones went on to answer the sub-questions in Q6 asking about the features of these courses, including:

- Whether the course is mandatory
- Course duration
- Industry involvement, that is internal or external/industry based
- Team size
- Team member and project allocation process
- Student team support mechanisms
- Assessment approaches

CAPSTONE PROJECT CHARACTERISTICS

MANDATORY OR NOT

In all cases the capstone projects formed a mandatory component of the degree. Two respondents noted that their capstones were only mandatory for students majoring in software engineering. Australian respondents mentioned that for them, capstones are a core component of their bachelor degree program’s ACS accreditation.

COURSE/PROJECT DURATION

Course duration (see Figure 6) varied from 10 weeks (a quarter) to year-long offerings (10-12 months). In this analysis, we have assumed that a semester runs between 12 and 16 weeks.

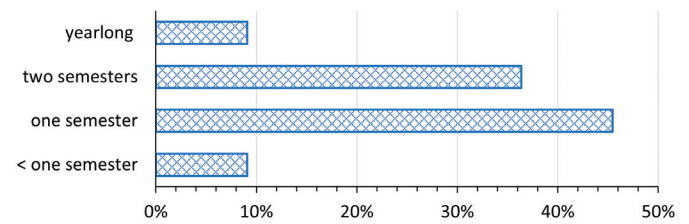


Figure 6: Capstone course duration (N = 26).

DEGREE OF INDUSTRY INVOLVEMENT

Only 15% of the respondents’ capstone projects involved clients who were external to their institution. Another 4% of projects were reported to be run as internal projects but had an industry link. Most capstone projects (60%) were conceived and delivered in-house. The remaining capstones had a combination of both internal and external projects/clients. One respondent noted that around 80% of their projects are normally internal. Another respondent mentioned that while all their projects are now internal, their institution used to offer external capstone projects. No reason was given for this change.

Team size: Most of the capstones surveyed involved teams composed of three to five students. The largest team size reported was seven.

Table 4: Typical team size.

Number of students	1-1	2-4	3-4	4-4	2-5	4-5	7-1
Percentage responses	32%	5%	26%	5%	11%	16%	5%

TEAM FORMATION

The most frequent mode of team formation was self-selection, where the responsibility for forming a team lay with the students. However, respondents noted that the project- and team-allocation process was facilitated by the course instructor’s institution when students either had difficulty finding a team or were latecomers. There were only two cases reported where the instructors always

allocated students to teams and projects. In one instance, the clients selected the students for their project from a pool of students who expressed interest in participating in their project. In this case, the instructor again dealt with the allocation of latecomers. Thirty-two percent of the courses involved projects undertaken by individual students. In these offerings, students selected a mentor and worked with them to determine a topic; in one case, the topic for each individual project was approved by an academic panel.

TEAM SUPPORT MECHANISMS

Four respondents noted that no specific team support mechanisms were used in their capstone projects; of these, one mentioned they used scaffolding involving multiple milestones and deliverables. Other respondents also felt that such scaffolds were necessary along with a clearly communicated set of expectations. Two respondents mentioned using a structured approach supported by agile processes and scrum as a mechanism for monitoring team progress and providing a framework for project delivery. Most of the respondents' courses leveraged experts to guide students during their projects. The terms academic mentor, supervisor, and coach were used to describe this role. In some cases, the course instructor mentored all project teams. In such cases, the instructor noted that specific in-class time and support for project work and team collaboration was needed. In cases where the teams are mentored by a large pool of academics, respondents highlighted the need for weekly or bi-weekly meetings or workshops involving all the students in the course. In other cases, each team was provided with both an academic mentor and an industry mentor. The industry mentor served as a subject-matter expert in terms of application domain, project management, and/or technical aspects of the project. The importance of the support and involvement of other project stakeholders,

not just the clients and academic mentors, was noted by one respondent. Some instructors noted mechanisms to facilitate inaugural client-team meetings, such as letters of introduction to the project site/client and academic team mentors accompanying their teams. Other mechanisms of support for teams and students involved in capstone projects included: space, equipment, insurance, contracting support, tendering support, access to venture capital, travel support, a portfolio team, and a careers team. Two respondents noted that their institution provided teams with financial support, but no further details were offered about how funds were allocated or could be used. The diversity of measures our respondents have in place highlights the complexity and level of support required to conduct capstone projects with industry partners.

ASSESSMENT STRATEGIES

Figure 7 illustrates the variety of reported capstone assessment approaches. This does not illustrate the combinations of assessment items in a single program but does illustrate the overall occurrences. For example, one approach involved the marking of six milestone documents, three incremental prototypes, and client feedback. Another instructor mentioned marking the product (technical artifacts) and practices (for example, project management methods) for all teams. Only two respondents used assessments that together provided a 360-degree evaluation [51] consisting of feedback from peers, clients, mentors, and instructors. One instructor highlighted the challenges that exist in fairly assessing student and team performance in a capstone course (this idea did not appear when we asked about the barriers and challenges in delivering computing capstone courses):

“... there are still gaps in assessing individual efforts and contributions and maintaining a consistent assessment method for projects that can vary widely in scope and technical depth.”

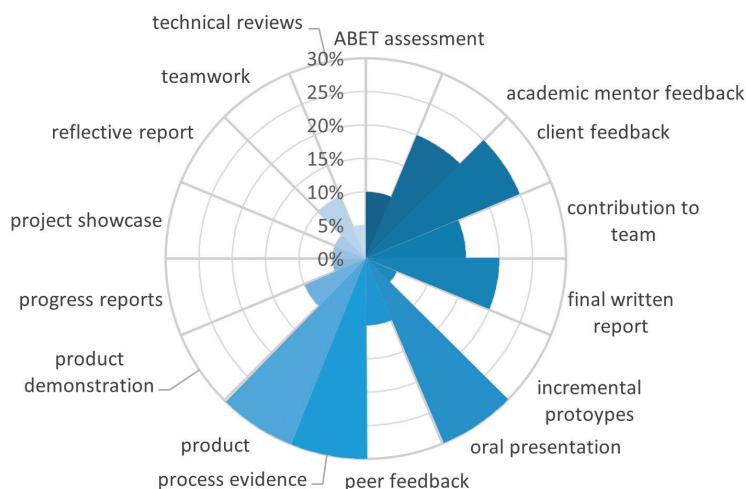


Figure 7: Assessment approaches (N = 26).

CHALLENGES AND BARRIERS

Twenty-six of the participants responded to the open question (Q4) about challenges and barriers to delivering a capstone project. Each response was allocated one or more codes from which the following seven themes emerged that are presented in the following subsections.

STUDENT PREPAREDNESS

From year-to-year, student cohort preparedness and engagement varies, which poses challenges when sourcing suitable projects that can cater to a diverse group of students. When faced with a large class, the range of capabilities and preparedness was noted as a significant challenge. This has implications for the level of support needed by students and in terms of sourcing suitable projects (when this is managed by the academic(s)) that are achievable and interesting to students. One participant noted that this constant change in cohort make-up poses challenges in their industry partners' experience.

"Predictability of size of student cohorts, demographic makeup of cohorts, their level of engagement, and technical and natural language competency. Dynamism here is challenging to manage, as the industry partner's experience ranges substantially between engagements."

STUDENT COMMITMENT

The theme of student commitment encompasses the idea of the level of commitment students must make to a capstone course and how this commitment can be affected by engagement and external factors.

"We recognize that some of our students have very busy schedules outside of classes so we ... use our class time to support project work and team collaboration."

"Many of our students work almost full time to support their studies. In addition, they are studying other courses that require teamwork in parallel with their capstone projects. This poses a challenge for our students. This workload can lead to disengagement for some students. Often the projects are carried out by the more engaged students, but these students can be the most time poor. An example of this is where one student led and took the load for a team while working part time with a small child at home and a working partner. Some students are not self-driven and rely on their teammates to manage them."

ACADEMIC COMMITMENT

This theme encompasses comments related to the commitment required and made by academics. It was a prevalent theme and discussed in all aspects of a program's delivery, from the time commitment to students/teams and maintaining industry relationships to the difficulty involved in keeping up with industry's rapid changes. In

Capstones, like many computing courses, must respond to rapid changes in technologies. When the students are embedded in industry projects this raises the stakes, and one academic noted that "The demands and speed of industry do not easily integrate into academic curriculum."

large-class capstone courses, unique challenges emerge that place demands on the academics coordinating the courses. For capstones that are run with industry clients, one respondent noted, *"It takes multiple full-time staff members to coordinate between students and employers."* Capstones, like many computing courses, must respond to rapid changes in technologies. When the students are embedded in industry projects, this raises the stakes, and one academic noted that *"The demands and speed of industry do not easily integrate into academic curriculum."* Also noted as a barrier was *"...finding the time ... to coordinate experiences between students and organizations."* This suggests that ultimately the level of commitment and effort for academics comes down to the high degree of coordination and logistical aspects of running a capstone course. In one capstone course, in which student teams are each given an academic mentor, differences in the level of engagement of mentors was noted:

"Team experiences vary. Some project supervisors support their teams through weekly meetings while others may only meet with their teams a few times throughout the year. Teams with less hands-on supervisors tend to become disengaged or flounder with poor processes and products being delivered. This of course has implications when it comes to client satisfaction."

SOURCING PROJECTS

A common theme related to challenges in sourcing projects and building relationships with industry partners was one contributing factor that emerged. These relationships are necessarily built over time and require a level of trust and commitment from the academics and industry clients involved. Moreover, sustaining these relationships over time takes effort and clear communication between both partners.

"It is labour intensive to develop, implement, and sustain ... work-based/project-based learning opportunities (building relationships, sourcing a sufficient number of appropriate projects)"

"It is hard to make community and industry contacts."

EXPECTATIONS

Another theme was related to a lack of alignment between the industry expectations and academic requirements. Two different challenges emerged; the first was with less 'technical' clients who do not have a realistic view of what can be achieved by students.

“The client is often not very technical. This can lead to unreasonable expectations being laced on the students.”

For students who have highly technical clients, the same demands can be made:

“One of our teams communicated with a developer as their point of contact in a client’s company. They expected the students to be able to learn new technologies and circumvent process in order to build the product. This led to conflict between the academic requirements of the course and the client’s demands.”

INDUSTRY CONTRIBUTION AND BUY-IN

One common pattern that emerged from the academics surveyed was the need for consistent commitment to the students.

“Employers must be willing to be involved in an ongoing commitment to the student.”

“We have found that at times the role of client is devolved to a junior staff member within the company which affects the student experience. The point of contact can be non-responsive and delay a team’s progress or be unclear on the project and its requirements leading to a lack of understanding of the requirements, poor team performance and ultimately a less than optimal outcome.”

RESOURCING

This theme encompasses challenges in staffing capstone courses and issues of lack of resources in terms of budget, space, equipment, and so on.

“Each project is different often requiring different technologies and methods which make it difficult for one instructor to be a subject matter expert across all projects.”

“It takes multiple full-time staff members to coordinate between students and industry partners.”

“... we have a limited budget which goes into our final public showcase. Some projects need specialist hardware and/or software that we simply do not have available for the students. Many industry clients will support the teams in providing access to these resources but [for] some this is not always possible. This means either the projects must be re-envisioned, or we have [to] say no to a potential industry collaboration.”

Discussion

This multi-institutional multinational (MIMN) analysis of computing instructor experiences has expanded the typical single institution perspective on capstone courses found in the computer science education research literature. That limited historical focus does raise the question of the consistency of research into capstone courses and the sustainability of innovations championed by solo enthusiasts [4,28]. We have found the empirical findings reported here map closely to the taxonomy of Tenhunen et al. [46], which was derived from the literature. Common elements of duration, team size, and student assessment exist. Tenhunen et al. also included *client and project sources*, which equated to our "industry involvement", and their *project implementation* attribute maps onto our "models of authentic learning and WiL." This confirms our view that the characteristics and themes captured by our study are key aspects of a classic capstone computing course.

In mapping the landscape of “capstone” courses in the broader context of work-integrated learning, we found that various models existed (see Figures 3 and 4). The most prevalent models reported in a 2012 survey of WiL opportunities in information and communications technology (ICT) in Australian universities were paid industry placements and industry-linked capstones. Less than 14% of opportunities were unpaid internships [43]. In contrast, our survey found that, in the context of computer science and information technology degrees, the most prevalent WiL opportunity was unpaid internships.

Systematic surveys of the literature report that most capstone projects run over one semester [20, 46]. Our findings suggest that, in practice, longer courses are more represented, with 45% of our respondents’ courses running for two semesters or more (aligning with the ACM/IEEE Software Engineering guidelines [3]—the Computer Science guidelines are more permissive on duration [2]). Tenhunen et al. [46] reported a greater variance in team sizes, from individual projects to teams of 35 students.¹ They also reported that most teams comprised four to five students. In contrast, we found a third of courses in our survey only offered individual projects and most were in-house with internal clients. All but one of the single-semester project courses were internal. Such projects give students an incorrect impression of the real world and forgo important professional skills, such as leadership and collaboration [49] in favor of a lower risk model. This prevalence of in-house projects with an internal client aligns with the courses described in the literature [46]. A 2023 article detailed an in-house scale capstone course where the project was run using a software house

¹ Further investigation of the source literature reporting large teams revealed that in such cases, the students operated within smaller sub-teams that worked on different aspects of the same project.

model in which students were allocated into one of four companies consisting of 20 students [31]. Each company had one student acting as scrum ‘lord’ who oversaw the project planning, processes, and company culture. Each sub-team within the company had a scrum leader. The authors argued that this model provides essential skills in multi-team software development processes, including coordination, collaboration, and integration skills that employers now demand. Thus, in the literature on capstone course curricula, it appears that a new trend toward running large-scale in-house capstone projects is beginning to emerge—echoing earlier models such as the “Kent IT Clinic” [16], where students worked in a managed simulation of an industry consulting service, but in that case serving external clients. Such initiatives are motivated in terms of building graduate capabilities, but it is also likely they might overcome the challenges faced in terms of sourcing industry projects and growing cohorts, which necessitate practical approaches to delivering capstone courses at scale with limited resources. While the initial set-up time of an internal software house model would be high, it may provide, over time, a low-cost means of coping with projects at scale without the risk of exposing students not ready for industry to the real world.

Some challenging real-world ‘clinic’ models integrate some aspects of the “Kent IT clinic” [16], but in a more supported connection with external clients. The Harvey Mudd CS Clinic model (<https://www.hmc.edu/cs/clinic/>) spawned from its engineering clinic [11], was mentioned to us by one reviewer. The Harvey Mudd Clinic was built on six essential characteristics of an engineering clinic, as presented in Bright and Phillips [11]:

1. *A teaching clinic, providing professional services.*
2. *Services meet professional standards.*
3. *Student participates under faculty direction.*
4. *Student’s participation is organized for learning as well as providing services.*
5. *Seminars or discussions are used to identify principles and procedures.*
6. *Student is given increasing responsibility.*

We have chosen to include this model, not only because a reviewer brought it to our attention, but also based on its commitment to a work-integrated model of learning, its characteristics, and longevity. Moreover, it offers a close match to the model we have been operating in our own university for several years, as the program has grown over the last two decades. While not established as a formal “clinic,” our university’s computing capstone course, as a critical aspect of the program, has its own character, folklore, and identity: Much like the Harvey Mudd clinic, it has evolved from a single coordinator to a more supported infrastructure. However, unlike the Harvey Mudd model,

we do not charge our clients for the service. We provide the service with no warranty and a disclaimer: “*There is therefore no guarantee that students will succeed in their efforts. This inherently means that the client assumes a degree of risk. This is part of an arrangement, which is intended to be of mutual benefit.*” It is worth noting, however, that as our student cohort has grown, we are suffering from many of the issues identified in our survey.

Student preparedness for capstone courses was an issue highlighted in our analysis; this was also discussed by Wright [49] more than 10 years ago. One proposed solution is to restructure the curricula and deliver a two-year holistic capstone experience [36]. In this model a pre-requisite project course is delivered in-house, building the skills and capabilities students need to move into a final year-long external industry-based project. Another approach is to examine the pre-requisite courses to verify that the assumed prerequisite skills and knowledge are indeed taught and assessed in earlier courses. One study from 2017 did just this aiming to locate the source of missing student skills [23]. A misalignment between learning outcomes and assessment in pre-requisite courses as well as lack of assessment of technical report writing and oral presentation skills was discovered. Both problems could be easily fixed in earlier courses. We believe it might mitigate some student preparedness issues for capstone course instructors to conduct a similar analysis to verify that the assumed prerequisite skills and knowledge are indeed taught and assessed in earlier courses.

Limitations

This survey’s findings have been inevitably constrained by the range and perspectives of the respondents. As such, it may not capture the full range of models and experiences that could equate to a capstone course. Capstones supporting individual students, non-software intensive projects, and supervised research projects are only partly addressed by the findings presented here. However, given the broad range and variety of courses our respondents have described, we believe this represents the reality of current practice delivering team-based software-intensive computing capstone courses in these contexts.

We did not ask respondents how large their cohorts were. Therefore, we have no way of knowing how class size might affect the delivery approaches used. Moreover, we did not ask respondents about the computing subdisciplines in which their capstone course was situated. It is possible that certain project characteristics and course delivery approaches are more suited to some subdisciplines than others. Exploring the dimensions of cohort size and subdisciplines and their influence on capstone course delivery in practice would be an interesting avenue for future research.

There were 26 (of the original 54 respondents, and 36 WiL respondents) responses to the questions about classic capstone project characteristics. While these responses alone may not be sufficient to generalize, as discussed, our findings are supported by prior literature, giving confidence that our findings are meaningful. It is important to recall that these 21 responses represent 21 institutional offerings and that the responses represent the reality of capstone courses rather than the ideal model. It does not mean all capstones necessarily fit with these findings. For example, at our institution, all projects are team-based industry projects with industry clients and weekly meetings with academic mentors—thereby combining a number of the characteristics identified in this survey.

We used an inductive reflexive thematic analysis approach to identify the challenges and issues our participants experienced as instructors of capstone projects. This analysis method is a “theoretically flexible interpretative approach to qualitative data analysis that facilitates the identification and analysis of patterns or themes in a given data set” [12]. This means there is always the risk of researcher bias, which was mitigated through two of the authors reviewing and revising themes with reference to the original data through a process of critique and discussion.

Our study was necessarily constrained by the stringent requirements imposed by our institutional review board (IRB) ethics consent process. We were unable to recruit participants individually by email, so contact had to be made through professional mailing lists, such as the SIGCSE mailing list. This makes it difficult to recruit participants even when a survey is brief and is one reason for the small sample size reported in this study. In addition, our IRB discourages the collection of demographic data in studies, even with the mandated informed consent process, where such data is viewed as not necessary to the research—our study fell into this category. This meant we were unable to follow up on responses to clarify a point or to provide demographic descriptions of the institutions. In the case of the open response questions, if there was any uncertainty or need for interpretation in a response or part thereof, then the portion that was ambiguous was not included in the analysis.

While the IRB and process are essential for ethical research and the process has benefits often leading to better research design, when the conditions imposed are too stringent and based on a medico-legal model of research, they impede research and innovation and can deter researchers from conducting certain studies, such as collaborative change-oriented studies involving action research [50]. Getting the ethical considerations for computer science education right and still being able to undertake human research that advances knowledge is critical for the future of high-quality computer science education research.

In this study we presented an overview of the reality of capstone courses and projects as seen by instructors of those offerings at a single point in time. We explored the capstone programs offered by our participants within the greater landscape of work integrated learning programs. Our analysis of computing capstone course characteristics, in practice, is supported by both the concurrent taxonomy developed by Tenhunen et al. [46], and the ACM/IEEE guides for capstone courses.

Conclusions

In this study, we presented an overview of the reality of capstone courses and projects as seen by instructors of those offerings at a single point in time. We explored the capstone programs offered by our participants within the greater landscape of WiL programs. Our analysis of computing capstone course characteristics, in practice, is supported by both the concurrent taxonomy developed by Tenhunen et al. [46] and the ACM/IEEE guides for capstone courses [2,3]. The software development/software engineering and team-based emphasis of our own experiences and focus of the study, does however limit the analysis, when considering the broader range of individual student, supervised, and research-based capstone projects undertaken in different settings. But, the set of features identified, along with these themes addressing issues and challenges, provides two theme-derived frameworks for future capstone research (Figure 8). The first is confirmation and elaboration of a set of typical capstone project characteristics. These characteristics were then used, and can be used in the future, to examine the changing landscape of capstone courses through a comparison with prior literature. In future studies, we recommend seeking information about cohort size, not just team size, to evaluate how capstone characteristics, types of capstones, and challenges encountered change with scale. The second framework arises from our complementary empirical exploration of the challenges instructors face when delivering their capstone courses, using a thematic analysis approach. This analysis resulted in the definition of a set of themes for future analysis of capstone courses and could be useful in designing courses with a focus on mitigating these challenges within the design. In the discussions section, we highlight possible mitigation strategies, which are rooted in prior literature, in some of the issues highlighted. Many of these issues need further research to empirically evaluate different

mitigation strategies. Some issues and barriers are clearly exacerbated as cohorts become larger—sourcing external projects, managing expectations and relationships, and resourcing all become more challenging as class sizes increase. Some early work on harnessing technology to help in the management of large-class capstone projects is emerging as one way of managing scale such as Gan et al.'s preference for wikis over learning-management systems [26] and use of learning analytics for grading [42]. However, such approaches need careful consideration and further investigation.

More work is needed to find ways to address the evolving issues identified through our survey, especially given the growing complexities in delivering a computing capstone course due to growing numbers of students. One challenge will be in responding to the practical needs to compromise on an ideal position, while not losing the real value of capstones with real-world clients, student teams, and projects when dealing with these issues of scale. Other areas that warrant further investigation include:

- Remote students/teams, where students may not be onsite, as during the Covid-19 pandemic [13]
- Global teams, where students may work in cross institutional teams [14,15,40,41]
- Multi-disciplinary teams
- Hybrid teams whose members work partly on campus and partly at home [44]
- Teams working on a client's site

A plethora of further topics would be worth exploring in this established but under-investigated area of practice in the computing curriculum, through further MIMN surveys, and studies of capstone courses.

MORE ONLINE

To find supplementary information to this article, visit <https://dl.acm.org/doi/DOI10.1145/3715880>

REFERENCES

1. ACS. The Australian Computer Society Accreditation Manual 5.3: Volume 2: Accreditation Criteria. (2022); <https://bit.ly/4aUEt4N>
2. ACM/IEEE Joint Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. Association for Computing Machinery Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. (2013); <https://bit.ly/4jPJcZD>
3. ACM/IEEE Joint Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. (2014); <https://www.acm.org/binaries/content/assets/education/se2014.pdf>.
4. Bain, J. Introduction (to the special Issue on Evaluation). *Higher Education Research Development* 18, 2 (1999), 165-172.
5. Bastarrica, M.C., Perovich, D., and Samary, M.M. What can students get from a software engineering capstone course? In *Proceedings of the 2017 IEEE/ACM 39th Intern. Conf. on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, 137-145. doi:10.1109/ICSE-SEET.2017.15.
6. Braught, G. and Siddiqui, F. Factors affecting project selection in an open source capstone. In *Proceedings of the 27th ACM Conf. on Innovation and Technology in Computer Science Education 1 (Dublin, Ireland) (ITICSE '22)*. (New York, NY, USA: ACM, 2022), 58-364. doi:10.1145/3502718.3524760.
7. Braun, V. and Clarke, V. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77-101. doi:10.1191/1478088706qp0630a
8. Braun, V. and Clarke, V. Thematic analysis. *APA Handbook of Research Methods in Psychology (2nd. ed.)*, (APA, 2012), 57-71. doi:10.1037/13620-004
9. Braun, V. and Clarke, V. One size fits all? What counts as quality practice in (reflexive) thematic analysis? *Qualitative Research in Psychology* 18, 3 (2021), 328-352.
10. Bridgeman, N. Project success: Defining, designing, constructing and presenting a capstone project. *16th Annual NACCQ Conf.* (NACCQ, 2003).
11. Bright, A. and Phillips, J.R. The Harvey Mudd Engineering Clinic Past, Present, Future. *J. Engineering Education* 88, 2 (1999), 189-194.
12. Byrne, D. A worked example of Braun and Clarke's approach to reflexive thematic analysis. *Quality & Quantity* 56, (2022), 1391-1412. doi:10.1007/s11135-021-01182-y.
13. Clear, T. Loosening ties: Permanently virtual teams and the melting iceberg of relationship. *ACM Inroads* 12, 3 (2021), 6-8. doi:10.1145/3479419.
14. Clear, T. and Beecham, S. Global software engineering education practice continuum: Special issue. *ACM Transactions on Computing Education (TOCE)* 19, 2 (2019), Article 7. doi: 10.1145/3294011.
15. Clear, T. et al. Challenges and recommendations for the design and

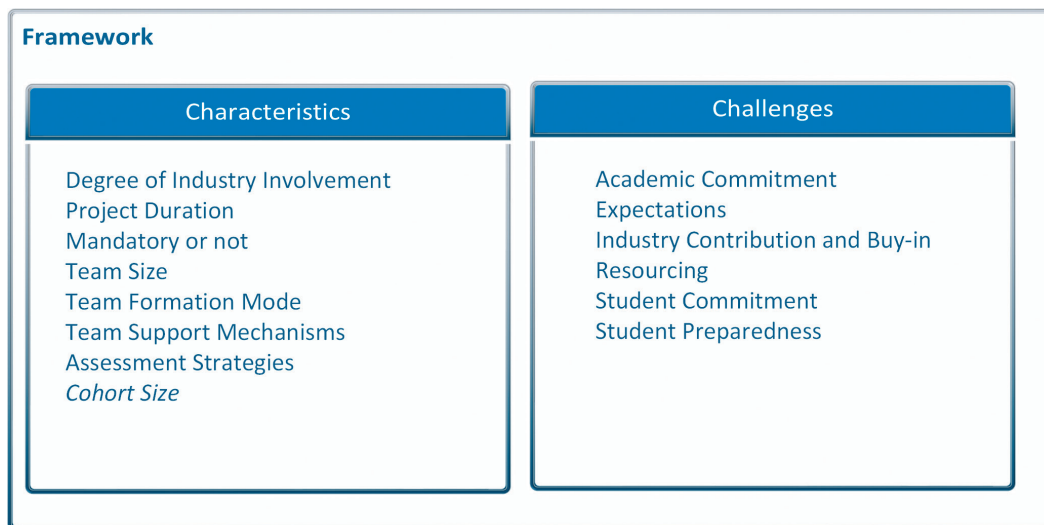


Figure 8: Assessment approaches. Note: Cohort size was identified as a gap in our original survey and is included here for completeness.

- conduct of global software engineering courses: A systematic review. In *Proceedings of the 2015 ITiCSE on Working Group Reports (ITiCSE-WGR '15)*. (New York, NY, USA: ACM, 2015), 1–39. doi:10.1145/2858796.285879.
16. Clear, T., Claxton, G. Thompson, S. and Fincher, S. Cooperative and work-integrated education in information technology. *International Handbook for Cooperative & Work-Integrated Education (2 ed.)*. Edited by R. Coll & K. Zegwaard. (2 ed.). (Lowell, MA, USA: World Association for Cooperative Education Inc, 2001), 141–150.
 17. Clear, T. et al. Resources for instructors of capstone courses in computing, in working group reports from ITiCSE on *Innovation and Technology in Computer Science Education (ITiCSE '01)*. (New York, NY, USA: ACM, 2001), 93–113. doi:10.1145/572133.572135.
 18. Conway, J.R., Lex, A., and Gehlenborg, N. UpSetR: An R package for the visualization of intersecting sets and their properties. *Bioinformatics* 33, 18 (2017), 2938–2940. doi:10.1093/bioinformatics/btx364, 2017.
 19. Ding, D., Yousef, M., and Yue, X. A case study for teaching students and agile and scrum in capstone course. *J. Computing Sciences in College* 32, 5 (Evansville, IN, USA: CCSC, 2017), 95–101. doi:10.5555/3069621.3069645.
 20. Dugan, R.F. A survey of computer science capstone course literature. *Computer Science Education* 21, 3 (2011), 201–267. doi:10.1080/08993408.2011.606118.
 21. Davis H.G. and Zilora, S.J. A tale of two capstones. In *Proceedings of the 17th Ann. Conf. on Information Technology Education (Boston, MA, USA) (SIGITE '16)*. (New York, NY, USA: ACM, 2016), 130–135. doi:10.1145/2978192.2978234.
 22. Deans, T. Service-learning in two keys: Paulo Freire's critical pedagogy in relation to John Dewey's pragmatism. *Michigan J. Community Service Learning*, (Fall 1999), 15–29.
 23. Elhussein, M. and Düşteğör, D. Tracing back the chain: Cognitive prerequisite analysis for CIS capstone project. In *Proceedings of the 9th Intern. Conf. on Education Technology and Computers (ICETC '17)*. (New York, NY, USA: ACM, 2017), 56–60. doi:10.1145/3175536.3175537.
 24. Fincher, S. Petre, M., and Clark, M. *Computer Science Project Work Principles and Pragmatics*. (London: Springer Verlag, 2001).
 25. Frezza, S. Clear, T., and Clear, A. Unpacking Dispositions in the CC2020 Computing Curriculum Overview Report. *50th ASEE/IEEE Frontiers in Education Conf.* (Uppsala, Sweden: IEEE, 2020).
 26. Gan B.K.S. and Ouh, E.L. Experience report on the use of technology to manage capstone course projects. *IEEE Frontiers in Education Conf. (FIE)*, (Uppsala, Sweden: IEEE, 2020), 1–8. doi:10.1109/FIE44824.2020.9274111.
 27. Goldweber, M. et al. A framework for enhancing the social good in computing education: A values approach. *ACM Inroads* 4, 1 (2013), 58–79.
 28. Gunn, C. They love it, but do they learn from it? Evaluating the educational impact of innovations. *Higher Education Research Development* 8, 2 (1999), 185–199.
 29. Hammer, D. and Berland, L.K. Confusing claims for data: A critique of common practices for presenting qualitative research on learning. *J. Learning Sciences* 23, 1 (2014), 37–46.
 30. Herrington, J., Oliver, R., and Reeves, T. Patterns of engagement in authentic online learning environments. In *Proceedings of the 19th Ann. Conf. of the Australasian Society for Computers in Learning in Tertiary Education, (ASCILITE)*, (Auckland, New Zealand: ASCILITE, 2002), 279–286.
 31. Li, Z.S., Arony, N.N., Devathanan, K., and Damian, D. Software is the easy part of software engineering—Lessons and experiences from a large-scale, multi-team capstone course. In *Proc. of the IEEE/ACM 45th Intern. Conf. on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. (2023), 223–234. doi:10.1109/ICSE-SEET58685.2023.00027.
 32. Lynch, K., Goold, A., and Blain, J. Students' pedagogical preferences in the delivery of IT capstone courses. *Issues in Informing Science and Information Technology* (2004), 431–442.
 33. Martin, N. Designing the IT capstone course: A systematic literature review. In *Proceedings of the 20th Ann. SIG Conf. Information Technology Education* (2019), 102–102.
 34. Mihás, P. Qualitative research methods: Approaches to qualitative data analysis. *Intern. Encyclopedia of Education*. 4th edition: edited by R.J. Tierney, F. Rizvi, and K. Ercikan. (Oxford: Elsevier, 2023), 302–313. doi:10.1016/B978-0-12-818630-5.11029-2
 35. Minnes, M., Serslev, S.G., and Padilla, O. What do CS students value in industry internships? *ACM Transactions on Computing Education (TOCE)* 21, 1 (2021), 1–15.
 36. Mohan, S., Chenoweth, S., and Bohner, S. Towards a better capstone experience. In *Proceedings of the 43rd ACM Technical Symp. Computer Science Education (SIGCSE '12)*. (New York, NY, USA: ACM, 2012), 111–116. doi:10.1145/2157136.2157173.
 37. Neyem, A., Benedetto, J.I., and Chacon, A.F. Improving software engineering education through an empirical approach: Lessons learned from capstone teaching experiences. In *Proceedings of the 45th ACM Technical Symp. on Computer Science Education (Atlanta, GA, USA) (SIGCSE'14)*. (New York, NY, USA: ACM, 2014), 391–396. doi:10.1145/2538862.2538920.
 38. Paasivaara, M. et al. How does participating in a capstone project with industrial customers affect student attitudes? In *Proceedings of the 40th Intern. Conf. on Software Engineering: Software Engineering Education and Training (Gothenburg, Sweden) (ICSE-SEET '18)*. (New York, NY, USA: ACM, 2018), 49–57. doi:10.1145/3183377.3183398.
 39. Patel, N., Brinkman, W-P., and Coughlan, J. Work placements and academic achievement: Undergraduate computing students. *Education+Training* 54, 6 (2012), 523–533.
 40. Pears, A. and Daniels, M. Developing global teamwork skills: The Runestone project. In *Proceedings of the IEEE EDUCON 2010 Conf.*, (2010), 1051–1056. doi:10.1109/EDUCON.2010.5492460
 41. Peters, A-K. et al. Preparing the global software engineer. In *Proceedings of the IEEE 10th Intern. Conf. on Global Software Engineering* (2015), 61–70. doi:10.1109/ICGSE.2015.20.
 42. Petkovic, D. Using learning analytics to assess capstone project teams. *Computer* 49, 1 (2016), 80–83. doi:10.1109/MC.2016.3.
 43. Pilgrim, C.J. and Koppi, T. Work-integrated learning rationale and practices in Australian information and communications technology degrees. In *Proceedings of the 14th Australasian Computing Education Conf. (ACE2012)*, (2012), 25–32. doi:10.5555/2483716.2483720.
 44. Smite, D., Christensen, E., Tell, P., and Russo, D. The future workplace: Characterizing the spectrum of hybrid work arrangements for software teams. *IEEE Software* 40, 2 (2023), 34–41. doi:10.1109/MS.2022.3230289.
 45. Tappert, C.C., Cotoranu, A., and Monaco, J.V. A real-world-projects capstone course in computing: A 15-year experience. In *Proceedings of the EDSIG Conf. on Information Systems and Computing Education*. (2015), <https://api.semanticscholar.org/CorpusID:114649782>.
 46. Tenhunen, S., Männistö, T., Luukkainen, M., and Ihtola, P. A systematic literature review of capstone courses in software engineering. *Information and Software Technology* 159 (2023). doi:10.1016/j.infsof.2023.107191.
 47. Vasankari, T. and Majanoja, A-M. Practical software engineering capstone course—Framework for large, open-ended projects to graduate student teams. *Computer Supported Education*. B. M. McLaren, R. Reilly, S. Zvacek, and J. Uholiohi (eds.). (Cham: Springer International Publishing, 2019), 310–327.
 48. Whalley, J. Goldweber, M., and Ogier, H. Student values and interests in capstone project selection. In *Proceedings of the 19th Australasian Computing Education Conf. (Geelong, VIC, Australia) (ACE '17)*. (New York, NY, USA: ACM, 2017), 90–94.
 49. Wright, K. Capstone programming courses considered harmful. *Commun. ACM* 53, 4 (2010), 124–127
 50. Zeni, J. A guide to ethical issues and action research. *Educational Action Research* 6, 1. (1998), 9–19.
 51. Zhang, J. et al. Toward next-generation engineering education: A case study of an engineering capstone project based on BIM technology in MEP systems. *Computer Applications in Engineering Education* 30, 1 (2022), 146–162.

JACQUELINE WHALLEY

Department of Computer and Information Sciences
Auckland University of Technology
Private Bag 92006
Auckland, 1142 New Zealand
jwhalley@aut.ac.nz

ASANTHIKA IMBULPITIYA

Information Technology Department
Otago Polytechnic Auckland International Campus
350 Queen Street
Auckland, 1010 New Zealand
Asanthika.Imbulpitiya@op.ac.nz

TONY CLEAR

Department of Computer and Information Sciences
Auckland University of Technology
Private Bag 92006
Auckland, 1142 New Zealand
Tony.Clear@aut.ac.nz

© 2025 Copyright held by owner/author(s). Publication rights licensed to ACM.

DOI: 10.1145/3715880