The Computing Dissertation: Lessons from a UK Practitioner Network

Steve Riddle*
Matthew Forshaw
steve.riddle@ncl.ac.uk
matthew.forshaw@ncl.ac.uk
Newcastle University
Newcastle upon Tyne, UK

Mark Zarb*
David Corsar
m.zarb@rgu.ac.uk
d.corsar1@rgu.ac.uk
Robert Gordon University
Aberdeen, Scotland, UK

Patrick Maier
Mario Kolberg
Hazrat Ali
patrick.maier@stir.ac.uk
mario.kolberg@stir.ac.uk
ali.hazrat@stir.ac.uk
University of Stirling
Stirling, Scotland, UK

Mohamed Elawady
Joseph El Gemayel
mohamed.elawady@strath.ac.uk
joseph.el-gemayel@strath.ac.uk
University of Strathclyde
Glasgow, Scotland, UK

Dharini Balasubramaniam
dharini@st-andrews.ac.uk
University of St Andrews
St Andrews, Scotland, UK

Hiroshi Shimodaira
h.shimodaira@ed.ac.uk
University of Edinburgh
Edinburgh, Scotland, UK

Miles Everett
miles.everett@abdn.ac.uk
University of Aberdeen
Aberdeen, Scotland, UK

Michael James Scott
michael.scott@falmouth.ac.uk
Falmouth University
Penryn, Cornwall, UK

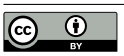
James Laird
j.d.laird@bath.ac.uk
University of Bath
Bath, UK

Abstract

The individual final-year project remains a core component of Computing degrees in the UK, at both undergraduate and postgraduate levels. It remains the main indicator of competency attained by a learner, providing evidence that they can successfully integrate knowledge from the different areas of their programme, master complexity and demonstrate metacognitive skills in reflecting on their own learning. However, growing class sizes, and the rise of generative AI raise questions about whether traditional dissertation-based assessment remains appropriate.

We report on a collaborative network of computing educators across UK higher education institutions, established to make practice in individual project modules explicit, visible, and open to peer critique through a Disciplinary Commons approach. Participants shared diverse approaches to assessment, supervision, moderation, and project allocation, while identifying common challenges around assessment consistency, academic integrity, ethics approval, and workload. Through structured peer exchange, the network surfaced practices considered valuable for wider adoption. We reflect on the outcomes of the structured peer sharing process, and identify directions for expanding the network and its outputs.

*Lead authors.



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGCSE Virtual 2026, Virtual Event, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2506-7/2026/11
https://doi.org/10.1145/3795867.3831035

CCS Concepts

• **Applied computing** → **Education**; • **General and reference** → **Surveys and overviews**.

Keywords

Education, Project-based learning, Assessment, Authentic assessment, Dissertation

ACM Reference Format:
Steve Riddle, Mark Zarb, Matthew Forshaw, David Corsar, Patrick Maier, Mario Kolberg, Hazrat Ali, Mohamed Elawady, Joseph El Gemayel, Dharini Balasubramaniam, Hiroshi Shimodaira, Miles Everett, Michael James Scott, and James Laird. 2026. The Computing Dissertation: Lessons from a UK Practitioner Network. In *Proceedings of the 2nd ACM Virtual Global Computing Education Conference V.1 (SIGCSE Virtual 2026), November 12–15, 2026, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3795867.3831035

1 Introduction

Project-based learning remains a cornerstone of undergraduate and postgraduate Computing programmes. It is valued for its alignment with professional practice and for its role in supporting deeper learning through the management of complexity and uncertainty in an authentic setting. In particular, the "capstone" or final project provides a means for students to integrate technical knowledge gained throughout their degree, develop technical skills and competencies, and engage authentically with open-ended problems.

While "capstone" is often used to refer to an integrative team-based project, we are concerned with *individual computing projects*,

sometimes called "dissertation" or (in the undergraduate case) "honours" projects. These are standalone, credit-bearing modules embedded within national and international curriculum guidance, including the ACM Computer Science Curricula [21] and the British Computer Society Accreditation Guidelines [15], giving students the opportunity to demonstrate their ability to manage a significant piece of work, as well as synthesise ideas to develop an artefact. In undergraduate programmes these projects are undertaken during the final year, often (but not exclusively) alongside taught modules. In taught postgraduate programmes the final 2.5–3 months are usually dedicated to the dissertation project, on completion of taught modules. Students work independently but are normally allocated an academic supervisor to advise and guide them.

As Computing programmes evolve, it is appropriate to ask questions about the appropriateness of the individual project as currently implemented. One such question is *whether it continues to offer an experience that is both authentic and educationally appropriate for contemporary cohorts?* At the same time, the context in which projects are delivered has changed substantially. Growing cohort sizes place increasing pressure on supervision and assessment practices [9], raising concerns about staff workload, consistency, and sustainability. Generative AI tools introduce further challenges, particularly around academic integrity, authorship, and the assessment of individual student contribution. These issues intersect with longstanding concerns about student confidence and competence, including variability in preparedness for independent work, project scoping, and self-directed learning. Together, these factors prompt a re-examination of how individual projects are designed, supported, and assessed, and whether current practices remain fit for purpose.

This paper is the result of a collaborative bid that brought together educators from UK higher education institutions with a shared interest in the development and assessment of individual Computing projects. The gathering and discourse drew inspiration from the Disciplinary Commons approach [14]. The method can draw its origins to Shulman's [36] work on teaching as community property and their call to end pedagogical solitude. Shulman's [37] later work explored the anatomy of courses alongside Hutchings' [19] exploration of how it might be expressed as a portfolio. Tenenberg and Wang [38] highlighted that such portfolios offered a means for computer science educators to "open classroom doors to one another, to critically examine the teaching that [the educator community] are enacting and the learning that students are undertaking" (p. 142). Fincher et al [11] provide an overview of multi-institutional teaching communities in computing education, some of which analyse and reflect on teaching in ways that accord with the Disciplinary Commons approach. Notable examples were distributed across Sweden, Australia, and New Zealand. Tenenberg & Fincher and Tenenberg [13] describe a formal bootstrapping model and a theoretically-derived framework for evaluation, describing their own efforts to form a commons. Ball et al [4] also convened a commons to investigate how to teach database topics. Lancaster et al [23] convened a commons to investigate approaches to project-based learning. Morrison, Ni & Guzdial [29] and Sentance, Humphreys, and Dorling [34] both applied the concept of Disciplinary Commons to support high school computing teachers in the USA and UK, respectively. It has also been applied beyond computing, e.g. to explore first-year engineering programmes [28].

In this instance, the Disciplinary Commons focused on sharing and reflecting on practice in project management, supervision, and assessment. In light of the contemporary pressures summarised above, the aim was to identify where innovation in practice is desirable, and to surface examples of successful approaches that can be shared and adopted within Computing and related disciplines.

This paper contributes: (1) a cross-institutional account of how individual Computing projects are currently organised across ten UK HEIs; (2) a synthesis of shared tensions around assessment consistency, workload, ethics, student preparedness, and generative AI; and (3) a set of transferable practices that institutions can use to review and improve their own project modules.

1.1 Research Questions

To understand how individual Computing projects are currently managed and assessed in practice, and how educators are responding to these challenges, we investigated three research questions:

(1) What approaches to the management, supervision and assessment of individual computing projects did participants bring to the network, and what patterns emerged from sharing these?
(2) What challenges did participants identify, and what approaches have they adopted or are exploring in response?
(3) What practices did the network collectively identify as worth sharing with the broader computing education community?

The scope of computing projects encompasses undergraduate and postgraduate programmes, in generalist and specialised degrees in UK HEIs. This experience report foregrounds what the network produced through structured peer sharing: a collective mapping of practice, identification of shared challenges, and examples of approaches worth adopting more broadly.

This paper is structured as follows: Section 2 reviews related work on challenges and solutions in individual projects; Section 3 describes the methodology used to gather and compare examples; Section 4 presents our findings and patterns surfaced across practice; Section 5 discusses implications and emerging challenges, including generative AI; and Section 6 returns to the research questions and concludes with limitations and directions for future work.

2 Related research

Individual projects have long enabled students to demonstrate synthesis of knowledge across a programme of study, reinforced by professional standards bodies. Their educational value lies in engaging students with complex, ill-structured problems [12]. The 2023 revision of the Computer Science Curricula [21] reaffirms the role of projects in providing hands-on experience while developing professional competencies. This guidance provides a foundation for examining how individual projects can be adapted to remain authentic and effective under contemporary constraints.

Compared with the literature on team-based capstone and software engineering projects, recent work focusing specifically on individual dissertation-style Computing projects appears more limited. A landmark research project over 25 years ago [12] surveyed UK and overseas computing departments and documented common practices in project work, including allocation, assessment, and supervision. Case studies included composites representing individual projects, group projects, and different types of project such

as research, design-and-build, capstone and client-based. Since then, much of the literature has focused on group-based project work, although reviews of capstone projects [39] identify challenges and recommendations also relevant to individual projects.

A number of challenges emerge from multiple recent studies, including some that focus on team-based "capstone" projects. Many are common to all forms of project work, for example:

- Growing class sizes [18, 27, 35] impact workload for instructors and supervisors, as well as the level of engagement of students who find themselves being supervised as part of a group in order to manage numbers.
- Sourcing projects externally can improve authenticity but bring challenges of provenance (requiring checking the quality and suitability of the project) [18, 22, 27].
- The processes for allocation of projects to students and to supervisors can challenge perceived fairness and workload [18, 27, 30].
- Student motivation and engagement is affected by cohort size, and has a significant impact on student experience [30, 35].
- Addressing the market gap between projects and industry needs, to improve authenticity and reduce the adaptation time for students to transfer to working in companies [6, 24, 25, 32].

Many of the challenges associated with individual projects are widely shared rather than institution-specific. These common challenges provide important background for examining how institutions currently organise and assess individual Computing projects.

2.1 Proposed solutions

2.1.1 Vetting of projects. Though time-consuming, several studies stress the importance of thoroughly vetting project proposals to ensure feasibility for the project duration and students' skill set: Naeem et al. [30] introduced a mandatory approval process (by supervisor and second marker) for students' proposals. They credit this intervention with improving retention and pass rates. Kurkovsky et al. [22] advocate a structured interview process to screen software engineering projects proposed by external partners for feasibility along multiple dimensions, including a partner's understanding of their project's requirements and commitment to support. The up-front screening cost (borne by the course coordinator) is offset by a smoother project experience and drives longer-term engagement with desirable partners. Migut et al. [27] have course coordinators vet proposals submitted by supervisors for feasibility and openness. They follow a strict template including background, research questions, references, and a baseline plan (not disclosed to students) to execute the project.

2.1.2 Scaffolding of engagement. Several studies report on a rigid project timetable forcing engagement (supervisor/client meetings), typically biweekly for a year-long project [30] and more often for shorter projects [27, 35], leading to better communication and tracking of progress between student, supervisor and, where applicable, industry client. Shakil and Denny[35] organise weekly progress meetings like stand-ups, incentivising engagement and opportunities for formative feedback.

2.1.3 Group/team supervision. Group supervision is a natural response to growing class sizes. Migut et al. [27] scaled their 10-week full-time (40h/week) research project course to classes of approximately 400 students by grouping students on individual but closely related projects into groups of 5. Groups were supervised by about 60 supervision teams, each consisting of one academic staff and at least one PhD student or postdoc, to lighten the staff workload and improve student access to supervisors. Fernandez [10] describes an undergraduate research experience module in which students form special-interest groups to improve peer support after submitting their initial project proposal.

2.1.4 Graded reflections. Reflective practice is known to strengthen unstructured learning from experience [5], e.g. while working on an individual project. Andrei et al. [2] include reflective essays in the graded assessments of all workplace projects (including the final dissertation project) of a graduate apprenticeship programme, thereby training the students in reflective journaling practice from early on. In addition to supporting learning, the reflective essays prove a rich source of student feedback, which the study used to evaluate students' perception of alignment between university teaching and workplace expectations.

2.1.5 Automated support for allocation. The problem of matching students to supervisors has been recognised as one that can be addressed by some degree of automation. For example, Zemaityte and Terzić [40] describe a tool to match student interests with a suitable supervisor, based on keywords and initial project proposal, with a high degree of successful allocation. More generally, the allocation problem has been studied across disciplines for many years, with linear programming solutions showing successful implementation in Engineering projects [33].

2.1.6 Other noteworthy recommendations. Shakil and Denny [35] motivate students at the start of their project by explicitly linking the skills mentioned in contemporaneous job ads to course's learning outcomes. They also address the staff experience in coordinating a capstone course by recommending to build a supportive community of instructors — which is also the aim of this project.

This literature review suggests that successful provision often involves introducing additional structure, e.g. clearer project scoping, regular checkpoints, and assessed interim activities. While these approaches can support engagement and feasibility, they also introduce trade-offs in terms of staff workload and student autonomy. These tensions are reflected in the practices reported in this study.

3 Methodology

In setting out to gather examples of practice from educators with an interest in individual capstone projects, we were inspired by the Disciplinary Commons approach [14]. *Disciplinary Commons* brings together educators within a shared discipline to make their teaching practices explicit, visible, and open to critique. Participants engage in structured cycles of reflection, documentation, and discussion, with a particular emphasis on the creation of artefacts that collectively form a public portfolio around a common pedagogic theme. Central to the approach is a peer review process, through which participants interrogate one another's practices, assumptions, and constraints, supporting both individual reflection and the development of shared disciplinary understanding.

We adopted a flexible adaptation of the Disciplinary Commons approach, retaining its emphasis on peer engagement and reflective sharing while using a less formal initial output than a public portfolio. While the original Disciplinary Commons model emphasises in-person meetings [14], our collaboration was conducted through a series of online meetings across the duration of the project, reflecting now-established norms for academic collaboration and enabling participation across institutions. This approach preserved the dialogic, peer-led character of the Disciplinary Commons while supporting sustained engagement among geographically distributed contributors. We started with an initial in-person meeting for collaborators to get to know each other, introduce their practice, and feel more comfortable in subsequent online meetings. We reserved a three-month period in the project duration for participants to hold their online meetings and discuss details about their institutions' project implementation. An end-of-project meeting was held as a hybrid event, opened up to more participants to discuss project outcomes.

Participants were recruited from the UK higher education community, as part of a 12-month funded project, via emails to relevant computing education mailing lists. We recruited thirteen participants from 10 UK Higher Education Institutions; a self-selecting sample of practitioners with a shared interest in individual projects, whose experiences and practices are the focus of this paper rather than a representative picture of UK HEI practice. It is important to note that whilst final-year projects are a mandatory part of the curriculum, it is typically up to each individual institution how to integrate this within their curricula.

We brought participants together for a one-day meeting at which everyone was invited to share their institutional practice in individual projects in the form of a 15-20 minute presentation. It was soon apparent that there was a large amount of variation in practice, including credit weighting, number and type of assessments, approaches to supervision and assessment/moderation. An initial summary of these elements is shown in Table 1.

The number of credits allocated to individual project modules vary from 30 to 60. One institution offers multiple versions of the project carrying different credits depending on the nature (single vs joint honours) and the duration or level of the programme. Project modules run over one or both semesters with one institution requiring preparation in semester one and project work in semester two. Cohort sizes varied from 30 to 350. Most institutions (9/10) limit the length of the final report, ranging from 12 pages to 15,000 words. Typically higher-level projects stipulate shorter reports, with specialist MSc courses stipulating short "research paper" reports which synthesis of the key results.

Following initial discussions, a period of structured peer discussion was established to enable participants to articulate and explore their project practices in greater depth. Institutions were initially grouped according to the credit value of the individual project, as this was taken to indicate a broadly comparable scale and level of student work. This resulted in a small number of peer groups, within which participants used a structured template to guide their discussion of their practice, key challenges encountered, and approaches adopted in response. This helped support cross-group comparison.

Table 1: Comparing Dissertation Modules by Institution

Module Properties			Assessment Methods (● Used or ○ Absent)									Report Limits	
Institution	Credits	Terms	Report	Artefact	Demo	Proposal	Ethics	Poster	Viva	Review	Recording	Word Count	Page Count
A	30	2	●	●	●	○	○	○	○	○	○	–	–
B	†	†	●	○	●	○	○	○	○	○	○	10,000	–
C	40	2	●	○	○	●	●	○	○	○	○	–	12
D	40	2	●	○	○	○	○	○	○	○	○	–	40
E	‡	‡	●	○	○	○	○	●	●	○	○	12,000	–
F	45	1	●	○	●	○	○	●	●	○	○	–	50
G	60	1	●	○	○	●	○	●	○	●	○	15,000	–
H	60	1	●	○	○	○	○	○	○	○	●	8,000	–
I	60	2	●	○	●	○	○	○	○	●	○	15,000	–
J	60	1	●	○	○	○	○	○	○	○	●	10,000	–

† Mixed model with different modular structures. ‡ 40 or a 60 credit variants.

Outcomes from peer-sharing were subsequently reviewed in an online meeting comprising of the full network, focusing on collectively identifying practices that were considered noteworthy, transferable, or illustrative of broader challenges in the design, supervision, and assessment of projects.

4 Findings

In this section we describe findings resulting from the paired discussions and other artefacts generated, in a process outlined below. An analysis of metrics for common elements across the projects is identified, followed by a presentation of themes broken into similarities, variations, challenges and trade-offs.

We considered the following sources: *a)* slide decks of presentations at the initial meeting, *b)* summary spreadsheet with details from all participating institutions, *c)* notes from paired discussions.

This discussion surfaced some patterns and points of difference across these sources, which are summarised in the themes below.

4.1 Patterns across Shared Practice

4.1.1 Common features of project organisation. There was a high degree of similarity in how projects are organised. In most cases, staff provide either a set of themes or a list of predefined project ideas, while also allowing students to propose their own projects. Only two institutions departed from this model, requiring students instead to select a topic that met specified constraints or aligned with the research interests of available supervisors. All institutions offer regular supervision, delivered either on an individual basis or in small groups. Weekly meetings are the norm, although some variation in frequency was reported. In terms of assessment, the final project report or dissertation is always the primary assessed component, and the highest-weighted. This suggests that, despite increasing interest in authentic and process-oriented assessment, the dissertation report remains the dominant proxy for judging independent project capability. This creates a tension between the holistic nature of project work and the narrowness of the principal assessment artefact. Some institutions also allocate credit to additional elements, such as interim progress submissions, software artefacts, presentations, demonstrations, or videos. In all cases,

projects are marked by at least two members of staff, with varying (but standard) mechanisms in place to moderate disagreements. These commonalities provided a baseline from which participants could identify and discuss more substantive points of difference.

4.1.2 Points of variation. Participants highlighted a number of areas of variation:

Skills Training: At the initial meeting, participants noted uneven levels of student preparedness in areas such as literature searching, academic writing, experimental design, data analysis, presentation skills and project management. Three institutions offer dedicated skills training within project modules to address these gaps. One institution had previously embedded such training within a *Professional and Social Aspects of Computing* module, but had withdrawn it due to low student engagement.

Attendance and engagement: Attendance and progress monitoring practices differed, with some institutions formally recording supervision attendance and issuing academic alerts for non-engagement, while others rely on more informal monitoring. The use of assessed interim deliverables was seen as one mechanism for encouraging sustained engagement. While all institutions assess a substantial final submission, some additionally assess (either formatively or summatively) interim deliverables or final presentations.

Identifying and addressing the use of generative AI: We acknowledge that this is changing at such a high rate that this information may be outdated as we write it. Institutional policies on good academic practice have been evolving in response to generative AI, with, at the time of writing, one institution requiring students to explicitly declare the extent of AI use using the AIAS scale [31]. Most institutions specified a reliance on the judgement of supervisors and markers to identify potential misuse, with confirmed cases handled through existing academic misconduct procedures.

4.1.3 Challenges identified by participants. Participants surfaced several challenges across their institutions, which were not always visible until placed alongside other practice. A key concern is achieving consistency in assessment across a diverse range of project types, including theory-based and software development projects, and research-led and industry-led work, spanning a wide range of technical areas. Differences in staff expectations about what constitutes a suitable individual project were seen as contributing to this challenge. Institutions attempt to mitigate these issues through marking guidance, second marking and moderation, although participants reported considerable variation in the level of detail provided. In some cases, limited guidance on double-marking practices can result in final marks being derived from the mean of two marks, which may not always reflect the quality of the work, except where large discrepancies trigger additional moderation.

Maintaining academic integrity remains an ongoing concern, particularly in relation to the use of generative AI in both software development and report writing. This has emerged as a significant and evolving challenge [7] and is discussed further in Section 5.3. Workload was also highlighted as a major issue, particularly in the context of increasing student–staff ratios. In some institutions, supervision extends across the full academic year, and uneven demand for popular topics (such as AI and data analytics) can exacerbate workload imbalances if not carefully managed. While double marking supports consistency, it further increases staff workload. Group supervision meetings can offer efficiencies, although this introduces certain challenges [16], and some institutions provide guidance on the amount of formative feedback supervisors should offer prior to submission in order to manage expectations and workload.

Ethics approval conducted within a single semester was identified as a further challenge. More rigorous ethics approval processes for work involving humans, animals or sensitive data, while necessary, can create time pressures for projects that require early data collection or user studies. Participants reported mitigating this risk by setting earlier ethics deadlines and requiring students to include fallback plans in case approval is delayed or denied.

4.1.4 Workload trade-offs. Several trade-offs related to workload management were discussed. Institutions were roughly evenly split between centralised and decentralised approaches to project allocation. Centralised allocation, typically managed by a coordinator and sometimes supported by a tool, can help balance supervision loads, avoid first-come-first-served dynamics for popular projects, and provide an overall view of the process. However, as cohort sizes grow, the administrative burden on coordinators can become substantial. Decentralised approaches distribute this workload across staff, but may disadvantage quieter or less confident students and can result in some students struggling to secure a project. These trade-offs did not resolve into clear recommendations; rather, surfacing them was itself a valued outcome of peer discussion, enabling participants to see their local decisions in a wider context.

A related trade-off concerns the use of single versus multiple project coordinators. A single coordinator maintains oversight across programmes and levels and manage cross-cutting issues, but incurs significant year-round workload. Sharing responsibility across multiple coordinators can reduce individual burden, but risks fragmentation without frequent and effective communication. Finally, institutions differed in their use of individual versus group supervision meetings. Group meetings can reduce supervisory load and allow students to benefit from shared advice and peer learning, particularly around common technical issues. However, participants noted that such settings may make it harder for quieter students to raise concerns or discuss difficulties in depth.

5 Discussion

The primary output of the network-building project, beyond mapping shared practice, was the identification of approaches participants considered worth sharing and adopting. These are presented first, followed by extended discussion of two main challenges where network conversation was particularly rich.

5.1 Innovative practices in project management

Institutions have adopted diverse forms of showcase presentations by students at the end of the project to highlight their achievements and demonstrate academic integrity. These include videos, posters, individual or class-wide demos, vivas and question and answer sessions. Class-wide events, to which all staff and (if feasible) the external examiner are invited, are particularly appreciated by students as a celebration of their capstone work. The Covid-19 pandemic impacted many in-person end of project activities and deliverables, which had to be moved online or turned into digital versions. While some have returned to their original form, digital

formats such as online presentations or recorded demonstrations have been retained in some institutions.

Risk management strategies have been adopted to reduce the impact of external factors on student projects. For example, one institution requires fully-internal project plans as backup for proposals involving external organisations. Several institutions obtained generic ethics approval for strictly constrained user studies (online surveys collecting only non-personally identifiable data from internal participants) from the School ethics committee. Students can use this approval as the basis for backup data collection or evaluation in case of delays or rejections with their full ethics application.

The use of commercial or in-house developed tools have been adopted to reduce workload and facilitate coordination activities such as project allocation and assessment support. For example, one institution developed an in-house Microsoft PowerApps tool which is used to centralise the marking and feedback processes.

Research-led teaching in several institutions extends to individual project modules. Staff embed aspects of their current research within project briefs, promoting the awareness and development of research as well as technical skills. This also has the potential to interest students in further study beyond their current programme.

Weekly drop-in sessions for skills support are offered in some institutions, tailoring provision to the needs of students and overcoming potential resistance to formal skills training (Section 4.1.2).

Some institutions expand the formal taught sessions to include guest talks from individual research groups in the Computing department. These provide an opportunity for all students to understand specific challenges for research in application areas such as AI, Game Engineering or Computing Education, regardless of whether they are taking a project in that area.

5.2 Student-Supervisor-Project Allocation

Institutions adopt a range of allocation mechanisms with common approaches including ranked student preferences over predefined projects, supervisor selection based on academic suitability, or approaches that combine preference data with programme-level constraints. Allocation may be manual, supported by spreadsheets or bespoke software, or automated using matching algorithms.

The "stable marriage" or matching problem [1] is the basis of algorithms [26] used to address student-project allocation, and these are adopted in some of the collaborating institutions. At one institution, student–supervisor allocation is treated as a core operational process across both undergraduate and postgraduate taught programmes, with an emphasis on scalability, transparency, and consistency. In undergraduate programmes, allocation typically follows a structured workflow: students state a preference for a research theme. The number of supervision "slots" for each theme is determined by staff workload. Once allocated to a theme, topic selection is managed by the staff overseeing that theme.

Programme leaders face a number of challenges in overseeing this process, including handling incomplete or conflicting preferences, managing uneven demand for popular supervisors, and responding to late changes in staff availability. Practice at one collaborating institution is now moving towards constraint optimisation tools with a focus on fairness and transparency of outcomes.

5.3 Generative AI

A shared challenge is the rise of use of AI chatbots amongst students to prepare assessment submissions, including the dissertation. For example, a survey by the Higher Education Policy Institute (HEPI) reports that in 2024, the prevalence of generative AI adoption among students reached 92%, primarily for text synthesis [17]. Overreliance on AI is reducing self-reflection potentially marginalizing active pedagogical engagement in favour of passive technological mediation [20]. AI chatbots and Large Language Models are the modern reality of the context students will graduate into. Participants generally regarded total prohibition as increasingly difficult to sustain, particularly where generative AI tools are already embedded in contemporary software development practice. Automating the detection of AI content remains technically fallible and prone to high false-discovery rates [8]. Partial remedies to counter the use of AI in project reports include mandating submission of codes, and coupling of dissertation assessment with rigorous oral defense or viva voce examinations to authenticate student authorship, though high class sizes can make this impractical. Likewise, the project assessment could be split in multiple low-stake sessions spread over entire duration of the project. Within individual projects and the other aspects of a Computing degree, it is essential to support students to develop critical analysis skills and competencies in the use of generative AI [3]. This will enable project supervisors and assessors to encourage appropriate and ethical use of these powerful tools.

6 Conclusions

We have addressed the three questions that framed the network's work. The patterns in shared practice (RQ1) and the challenges participants identified (RQ2) are discussed in Sections 4.2.1–4.2.4. The practices the network considered worth sharing more widely (RQ3) are presented in Section 5.1. The contribution here is not a systematic account of UK practice as the self-selecting nature of the network precludes that, but a structured record of what a community of practitioners learned by making their practice visible.

We have seen that dissertations remain a central part of the assessment of individual projects, but their role is threatened by pressures of workload and capabilities of generative AI.

Beyond this project, there is more to do to fully address how innovative practices can be shared more widely, and to widen the scope of review to include student perspectives. A forthcoming framework, distilled from this and related work, aims to provide practitioners with a typology for situating their own institutional project model and a reflective tool for identifying areas for development. Future work will invite network participants to evaluate and refine this framework against their own experience, and move to extend, adapt and summarise these practices in a suitably accessible and digestible way.

Acknowledgments

The work reported in this paper was funded by a CPHC Special Project Grant, *Cultivating Collaborative Networks for Individual Computing Projects and Dissertations.*

References

[1] David J. Abraham, Robert W. Irving, and David F. Manlove. 2003. The Student-Project Allocation Problem. In *Algorithms and Computation: 14th International Symposium, ISAAC 2003*. 474–484.

[2] Oana Andrei, Matthew Barr, Syed Waqar Nabi, and Alistair Morrison. 2025. Understanding Skill Transfer Between University and Workplace Through Reflective Practice: A Software Engineering Work-Based Learning Experience. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1, ITiCSE 2025, Nijmegen, The Netherlands, 27 June 2025 - 2 July 2025*. ACM, 368–374. doi:10.1145/3724363.3729119

[3] Ravinithesh Annapureddy, Alessandro Fornaroli, and Daniel Gatica-Perez. 2025. Generative AI Literacy: Twelve Defining Competencies. *Digit. Gov.: Res. Pract.* 6, 1, Article 13 (Feb. 2025), 21 pages. doi:10.1145/3685680

[4] Les Ball, Sheila Baron, Charles Boisvert, Richard Cooper, Tugrul Essendal, Sally Fincher, Tony Jenkins, Petra Leimich, Al Monger, David Nelson, Thomas Neligwa, James Paterson, Clare Stanier, Tony Valsamidis, and John Wilson. 2010. A Disciplinary Commons for Database Teaching. In *Proceedings of the 8th International Workshop on Teaching, Learning, and Assessment of Databases*. 6–9.

[5] Evelyn M Boyd and Ann W Fales. 1983. Reflective learning: Key to learning from experience. *Journal of humanistic psychology* 23, 2 (1983), 99–117.

[6] Eleni Chatzidaki, Sofia Papavlasopoulou, Letizia Jaccheri, and Michail Giannakos. 2025. Identifying Project Management Skills in Industry-Driven Capstone Projects. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*. 959–969.

[7] Tony Clear, Åsa Cajander, Alison Clear, Roger McDermott, Mats Daniels, Monica Divitini, Matthew Forshaw, Niklas Humble, Maria Kasinidou, Styliani Kleanthous, et al. 2025. AI integration in the IT professional workplace: a scoping review and interview study with implications for education and professional competencies. *2024 Working Group Reports on Innovation and Technology in Computer Science Education* (2025), 34–67.

[8] Mauro Coccoli and Gabriella Patanè. 2024. AI vs. AI: The Detection Game. In *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*. IEEE, 1–6.

[9] César Domínguez, Arturo Jaime, Francisco J García-Izquierdo, and Juan J Olarte. 2020. Factors considered in the assessment of computer science engineering capstone projects and their influence on discrepancies between assessors. *ACM Transactions on Computing Education (TOCE)* 20, 2 (2020), 1–23.

[10] Amanda S. Fernandez. 2025. A Course-based Undergraduate Research Experience (CURE) Focused Broadly on Research Methods in Computer Science. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 332–338. doi:10.1145/3641554.3701805

[11] Sally Fincher, Raymond Lister, Arnold Pears, Judy Sheard, Josh Tenenberg, and Alison Young. 2006. Multi-institutional teaching communities in computer education. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*. 7–10.

[12] Sally Fincher, Marian Petre, and Martyn Clark (Eds.). 2001. *Computer Science Project Work: Principles and Pragmatics*. Springer. doi:10.1007/978-1-4471-3700-9

[13] Sally Fincher and Josh Tenenberg. 2006. Using theory to inform capacity-building: Bootstrapping communities of practice in computer science education research. *Journal of Engineering Education* 95, 4 (2006), 265–277.

[14] Sally Fincher and Josh Tenenberg. 2011. *A Commons Leader's Vade Mecum: How to instantiate a Disciplinary Commons*. University of Kent Press.

[15] BCS The Chartered Institute for IT. 2022. Academic Accreditation Guidelines. https://www.bcs.org/media/1209/accreditation-guidelines.pdf

[16] Matthew Forshaw, Cristina Adriana Alexandru, Caitlin Bentley, Vladimiro González-Zelaya, Vangel Ajanovski, Mireilla Bikanga Ada, Julian Brooks, Joshua Burridge, Alex Chao, Rutwa Engineer, et al. 2025. Fairness in Student Group Formation: Perspectives, Priorities, Compromises, Mechanisms, and Tooling. In *2025 Working Group Reports on Innovation and Technology in Computer Science Education*. 217–276.

[17] Josh Freeman. 2025. Student Generative AI Survey 2025. Retrieved 1 December 2025 from https://www.hepi.ac.uk/reports/student-generative-ai-survey-2025/

[18] Sara Hooshangi, Asma Shakil, Subhasish Dasgupta, Karen C. Davis, Mohammed Farghally, KellyAnn Fitzpatrick, Mirela Gutica, Ryan Hardt, Steve Riddle, and Mohammed Seyam. 2024. Instructors' Perspectives on Capstone Courses in Computing Fields: A Mixed-Methods Study. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education, ITiCSE 2024, Milan, Italy, 8 July 2024*, Mattia Monga, Violetta Lonati, Erik Barendsen, Judithe Sheard, and James H. Paterson (Eds.). ACM, 68–94. doi:10.1145/3689187.3709608

[19] Pat Hutchings. 1998. How to develop a course portfolio. In *The Course Portfolio*. Routledge, 47–75.

[20] Vitomir Kovanović, Abhinava Barthakur, Srećko Joksimović, and George Siemens. 2025. Why universities need to radically rethink exams in the age of AI. *Nature* 648, 8092 (2025), 35–37.

[21] Amruth N. Kumar and Rajendra K. Raj. 2024. Computer Science Curricula 2023 (CS2023): The Final Report. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 1867–1868. doi:10.1145/3626253.3633405

[22] Stan Kurkovsky, Chad A. Williams, Mikey Goldweber, and Nathan Sommer. 2024. External Projects and Partners: Addressing Challenges and Minimizing Risks from the Outset. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1, ITiCSE 2024, Milan, Italy, July 8-10, 2024*. ACM. doi:10.1145/3649217.3653593

[23] Thomas Lancaster, Tony Jenkins, Leonor Barroca, Maurice Calvert, Siobhan Devlin, Richard Foley, Julie Horton, John Moore, and Paula Sturdy. 2011. Some good ideas for student projects from the disciplinary commons. In *Proceedings of 12th Annual Conference of the Higher Education Academy Subject Centre for Information and Computer Sciences, Belfast*.

[24] Wendy A. Lawrence-Fowler, Laura M. Grabowski, and Christine F. Reilly. 2015. Bridging the divide: Strategies for college to career readiness in computer science. In *2015 IEEE Frontiers in Education Conference (FIE)*. 1–8. doi:10.1109/FIE.2015.7344317

[25] Gunhild M. Lundberg, Birgit R. Krogstie, and John Krogstie. 2020. Becoming Fully Operational: Employability and the Need for Training of Computer Science Graduates. In *2020 IEEE Global Engineering Education Conference (EDUCON)*. 644–651. doi:10.1109/EDUCON45650.2020.9125188

[26] David Manlove, Duncan Milne, and Sofiat Olaosebikan. 2022. Student-project allocation with preferences over projects: Algorithmic and experimental results. *Discrete Applied Mathematics* 308 (2022), 220–234. doi:10.1016/j.dam.2020.08.015 Combinatorial Optimization ISCO 2018.

[27] Gosia Migut, Aleksander Buszydlik, and Mathijs Michiel de Weerdt. 2025. The Research Project in Computer Science Bachelor Education: Undergraduate Research Experience at Scale. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1, ITiCSE 2025, Nijmegen, The Netherlands, 27 June 2025 - 2 July 2025*. ACM, 410–416. doi:10.1145/3724363.3729101

[28] M Jean Mohammadi-Aragh and Rachel L Kajfez. 2015. Comparing common and discipline-specific first-year engineering programs: Examining the need and explaining the study. In *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–4.

[29] Briana B Morrison, Lijun Ni, and Mark Guzdial. 2012. Adapting the disciplinary commons model for high school teachers: improving recruitment, creating community. In *Proceedings of the ninth annual international conference on International computing education research*. 47–54.

[30] Usman Naeem, Syed Islam, and Arish Siddiqui. 2019. Improving Student Engagement and Performance in Computing Final Year Projects. In *IEEE International Conference on Engineering, Technology and Education, TALE 2019, Yogyakarta, Indonesia, December 10-13, 2019*. IEEE, 1–9. doi:10.1109/TALE48000.2019.9225860

[31] Mike Perkins, Leon Furze, Jasper Roe, and Jason MacVaugh. 2024. The Artificial Intelligence Assessment Scale (AIAS): A Framework for Ethical Integration of Generative AI. *Journal of University Teaching and Learning Practice* 21, 06 (Apr. 2024). doi:10.53761/q3azde36

[32] Alex Radermacher, Gursimran Walia, and Dean Knudson. 2014. Investigating the skill gap between graduating students and industry expectations. In *Companion Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE Companion 2014)*. 291–300. doi:10.1145/2591062.2591159

[33] Johnson Ramotsisi, Mompoloki Kgomotso, and Lone Seboni. 2022. An Optimization Model for the Student-to-Project Supervisor Assignment Problem-The Case of an Engineering Department. *Journal of Optimization* 2022, 1 (2022), 9415210. doi:10.1155/2022/9415210

[34] Sue Sentance, Simon Humphreys, and Mark Dorling. 2014. The network of teaching excellence in computer science and master teachers. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. 80–88.

[35] Asma Shakil and Paul Denny. 2024. Enhancing Student Engagement in Large-Scale Capstone Courses: An Experience Report. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1, ITiCSE 2024, Milan, Italy, July 8-10, 2024*, Mattia Monga, Violetta Lonati, Erik Barendsen, Judithe Sheard, and James H. Paterson (Eds.). ACM. doi:10.1145/3649217.3653580

[36] Lee S Shulman. 1993. Teaching as community property: Putting an end to pedagogical solitude. *Change: The Magazine of Higher Learning* 25, 6 (1993), 6–7.

[37] Lee S Shulman. 1998. Course anatomy: The dissection and analysis of knowledge through teaching. In *The course portfolio*. Routledge, 5–12.

[38] Josh Tenenberg and Qi Wang. 2005. Using course portfolios to create a disciplinary commons across institutions. *Journal of Computing Sciences in Colleges* 21, 1 (2005), 142–149.

[39] Saara Tenhunen, Tomi Männistö, Matti Luukkainen, and Petri Ihantola. 2023. A systematic literature review of capstone courses in software engineering. *Inf. Softw. Technol.* 159 (2023), 107191. doi:10.1016/J.INFSOF.2023.107191

[40] Gintare Zemaityte and Kasim Terzić. 2019. Supervisor Recommendation Tool for Computer Science Projects. In *Proceedings of the 3rd Conference on Computing Education Practice (CEP '19)* (Durham, United Kingdom). Article 1, 4 pages. doi:10.1145/3294016.3294030