Embedding External Partner Code Reviews into Mobile Application Development Coursework

Lee Clift
lee.clift@strath.ac.uk
University of Strathclyde
Glasgow, Scotland

Donna Brawley
donna.brawley@strath.ac.uk
University of Strathclyde
Glasgow, Scotland

Robert Greer
robert.greer@strath.ac.uk
University of Strathclyde
Glasgow, Scotland

Damien Anderson
damien.anderson@strath.ac.uk
University of Strathclyde
Glasgow, Scotland

Alasdair Lambert
alasdair.lambert@strath.ac.uk
University of Strathclyde
Glasgow, Scotland

Abstract

Code review is a standard practice in professional software development, yet its integration into assessed coursework remains uncommon in computing higher education. This paper describes the design and implementation of fortnightly, industry-led code reviews as a core component of a mobile application development coursework, in which postgraduate students develop features for an existing application. We outline the assessment structure, the role of external senior developers as reviewers, and how feedback and marking are managed across the review cycle. Practical guidance is offered for educators looking to adopt a similar approach, covering partnerships, student preparation, and scheduling.

CCS Concepts

• **Social and professional topics** → *Software engineering education*; **Student assessment**.

Keywords

Education, Code Review, Software Engineering

ACM Reference Format:
Lee Clift, Donna Brawley, Robert Greer, Damien Anderson, and Alasdair Lambert. 2026. Embedding External Partner Code Reviews into Mobile Application Development Coursework. In *Proceedings of the 31st ACM Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2026), July 10–15, 2026, Madrid, Spain.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3803401.3811982

1 Introduction

Code reviews within software engineering are a structured process in which source code is examined by individuals other than the original author to identify defects, improve quality, and share knowledge across a team. In industry, it is considered a standard practice within collaborative software development, particularly in agile environments where iterative feedback is central to the development cycle [1].

This work is licensed under a Creative Commons Attribution 4.0 International License.
ITiCSE 2026, Madrid, Spain
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2633-0/2026/07
https://doi.org/10.1145/3803401.3811982

The use of code review within computing education is not entirely new. Prior work has explored peer code review as a formative tool [2], noting benefits for both reviewers and reviewees in terms of code quality awareness and collaborative skills. However, industry-facing code review, where students receive feedback from professional developers on a live codebase, remains less commonly reported in the literature.

This paper describes integrating structured, professional code reviews into a mobile application development course and offers practical guidance for educators looking to adopt a similar approach.

2 The Assessment and Code Reviews

2.1 The Cohort

The coursework is delivered to 30 taught postgraduate students in a mobile application development module as part of a Software Engineering degree program in the United Kingdom. Students work in teams of five, formed by self-selection, and have 14 weeks of programming experience on the programme before the coursework begins. Flutter and Dart are introduced for the first time in this module.

2.2 The Assessment Context

Rather than developing a standalone application from scratch, students implement new features for the university's own mobile application, a live product used by students and staff. Each team selects a feature from a set of proposals provided by the university's app development team, or proposes their own with instructor approval. Proposed features are fully designed in a prior coursework, keeping the focus of this assessment on development. Teams produce a standalone prototype suitable for presentation to the app team for potential inclusion in the live codebase.

2.3 The Review Process

Central to the assessment is a fortnightly code review cycle. Every two weeks, each team meets with a senior developer from the university's app development team for a structured 30-minute session. These are not simple progress checks; they are designed to mirror professional development practice.

Students arrive with working, committed code and lead the session themselves: walking the reviewer through their work, explaining design decisions, and responding to technical questions. A maximum of 3 additional questions for the reviewer can be submitted 48 hours in advance. This limitation is purposely provided to ensure teams think meaningfully about what they wish to ask the reviewers. The reviewer's role is to probe, challenge, and offer feedback, not to teach - all questions directed to reviewers should be technical, not assessment-related.

Because reviewers are not members of the teaching team, their feedback carries different weight. Students are presenting to a legitimate stakeholder, not performing for a marker. This produces more considered preparation and more meaningful responses to feedback. Students consistently report that positive feedback from a senior developer feels more authentic and validating than equivalent feedback from an academic.

2.4 Feedback and Iteration

Feedback is expected to be visibly addressed before the next session, creating an iterative loop throughout the coursework: build, review, reflect, improve, repeat. Students keep notes on feedback received and reference these in subsequent sessions, demonstrating understanding as well as change. Improvement across the process is explicitly factored into the final mark.

The fortnightly cadence also shapes working habits more broadly, discouraging late-stage rushing and cramming, in favour of incremental progress. This maps well onto agile practice and is useful preparation for the master's thesis that follows this module, which is similarly expected to be developed incrementally, as well as future job roles.

2.5 Assessment and Marking

Code reviews contribute to the assessment through a dedicated professionalism component, accounting for 20% of the total grade. This captures preparation, communication, and the ability to justify technical decisions - behaviours not visible in a typical final submission. After each session, reviewers provide a short written summary to the module coordinator, which forms the basis for this mark, alongside other measures, such as version control usage and general visibility during team meetings and dedicated labs.

Marking is based on trajectory rather than performance at any single point, which is communicated explicitly to students at the outset. Reviewer notes raising wider concerns, such as team dynamics issues, are flagged to the coordinator separately and handled through typical groupwork techniques.

3 Lessons Learned & Practical Guidance

3.1 Industry Involvement and Partnerships

The most immediate question for educators considering this approach is how to secure industry involvement. Alumni networks, industry advisory boards, and existing research or consultancy relationships are all viable routes, and in our experience partners are generally willing once the time commitment - one 30-minute session per fortnight - is made clear. In this implementation, involvement of the university app team has created a visible pipeline for internships and junior roles, with students observed working in an authentic environment before any formal hiring process begins.

There is also no requirement for the partner to be a commercial organisation. Departmental research groups, local charities, or community projects can all provide the same authentic context, provided there is a genuine stakeholder invested in the outcome.

Onboarding reviewers before the coursework begins is important. Without some framing, sessions can drift toward open-ended discussion rather than structured feedback. A brief set of focus areas ahead of each round and a short introductory meeting at the start of the coursework is sufficient to keep sessions consistent across teams.

3.2 Preparing Students

Students unfamiliar with professional code review can find the first session uncomfortable. A short preparatory session beforehand, covering what to expect and how to present work effectively, helps considerably. Having teams return to a communal area after their review is also useful - once the first group returns, the rest of the cohort can see the experience is less intimidating than anticipated, and in many cases, exciting and validating.

Students should also understand clearly what is being assessed: trajectory, communication, and responsiveness to feedback, not just code quality. With this framing, students treat the review as a formal meeting, and not simply an informal chat.

3.3 Logistics and Scheduling

Fortnightly reviews across a full cohort require fixed scheduling from the outset. Treating review slots as immovable, in the same way a sprint review would be in an agile team , prevents rescheduling from becoming a recurring overhead. Blocking reviewer time as a single fortnightly session split across teams, rather than individual appointments, makes the commitment manageable for external partners and easier to sustain across a full semester.

4 Conclusion

Embedding professional code reviews into assessed coursework is a relatively low-overhead change that yields meaningful returns: increased student engagement, professional skill development, and greater authenticity in the assessment experience. The approach described here is not dependent on a particular technology stack or institutional context; the core principles of rhythm, external audience, and iterative feedback are transferable across programmes and settings.

References

[1] Alberto Bacchelli and Christian Bird. 2013. Expectations, Outcomes, and Challenges of Modern Code Review. In *Proceedings of the 35th International Conference on Software Engineering (ICSE '13)*. IEEE, 712–721. doi:10.1109/ICSE.2013.6606617

[2] Theresia Devi Indriasari, Andrew Luxton-Reilly, and Paul Denny. 2020. A Review of Peer Code Review in Higher Education. *ACM Trans. Comput. Educ.* 20, 3, Article 22, 25 pages. doi:10.1145/3403935