Github Classroom: Student Attainment, Feedback, Feedforward and Industrial Practice

Sebastian Blair

University of Greenwich

Abstract

This review evaluates GitHub Classroom as a pedagogical tool in higher education, with reference to computing and digital engineering programmes. The platform facilitates authentic assessment, aligns teaching with industry-standard practices and provides opportunities for engagement, monitoring and feedback. While its strengths include transparency, scalability and formative assessment, challenges remain in relation to the necessary steep learning curve, institutional integration and equity of access. The review concludes with recommendations for practitioners and directions for future research, highlighting the need to balance professional authenticity with inclusive pedagogy.

Introduction

Version control systems (VCS) are fundamental to modern software development and collaborative digital work. Git, created by Linus Torvalds to manage the Linux kernel, has become the dominant system for managing source code and related artefacts (Chacon and Straub, 2014). GitHub, a widely used hosting platform, enables developers to collaborate on over 270 million repositories (GitHub, 2023).

In higher education (HE), teaching VCS is no longer optional. With the advent of generative AI tools capable of automating many coding tasks, the ability to manage, review and collaborate on code has arguably become as important as writing code itself (Fowler, 2023). GitHub Classroom (GHC) (GitHub, 2025a) extends GitHub's collaborative infrastructure to teaching contexts by enabling instructors to distribute assignments, manage repositories and review student submissions.

This review evaluates GHC as a tool for supporting student attainment, enabling feedback and feedforward and embedding industrial practice into HE curricula.

Applications of GitHub Classroom in teaching and learning

GHC (figure 1) extends the existing GitHub infrastructure by enabling instructors to create, distribute and manage assignments efficiently. On enrolment, each student – or team – receives a private repository that can be pre-populated with starter code or template files, providing a consistent foundation for the task. Students then complete their work using standard Git workflows ('add', 'commits', 'branches', 'merges', 'rebases' and 'pull requests'), thereby engaging directly with industry-relevant practices. Registration through institutional email addresses allows instructors reliably to associate repositories with individual students,

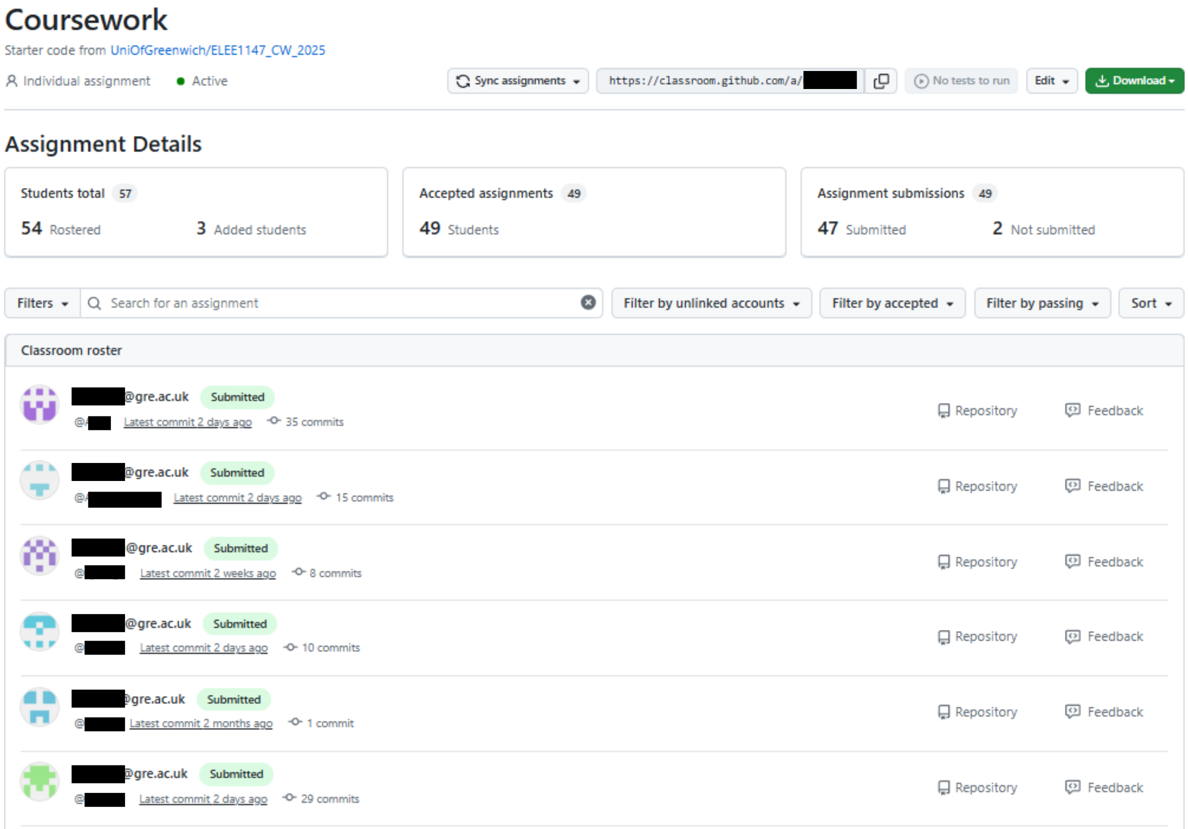

Figure 1. Screenshot of GHC assignment details

making it possible to identify non-participation or delayed engagement. In this way, GHC not only streamlines assignment management but also offers a mechanism for early intervention when patterns of disengagement emerge.

Pedagogical applications of GitHub in teaching are varied and extend well beyond simple version control. One significant advantage is the potential for authentic assessment, as students can work within the same tools and practices used in industry, so developing employability skills alongside their subject knowledge. The platform also facilitates feedback and feedforward, instructors being able to provide comments directly within repositories through pull requests, issues or inline annotations and thus ensuring that the feedback is both contextualised and actionable (figure 2).

In addition, GitHub offers opportunities for engagement tracking, since commit histories allow instructors to gain fine-grained insights into student activity and engagement patterns. This enables timely interventions when contributions appear sparse or absent. The platform equally supports collaborative learning, with group assignments enhanced by branching and merging workflows that encourage students to practise collaboration in a structured environment.

Finally, GitHub lends itself to scaffolded learning, as educators can design starter templates and incremental assignments that progressively build student competence not only in coding but also in VCS practices. This dual focus helps ensure that students gain both technical proficiency and professional readiness.

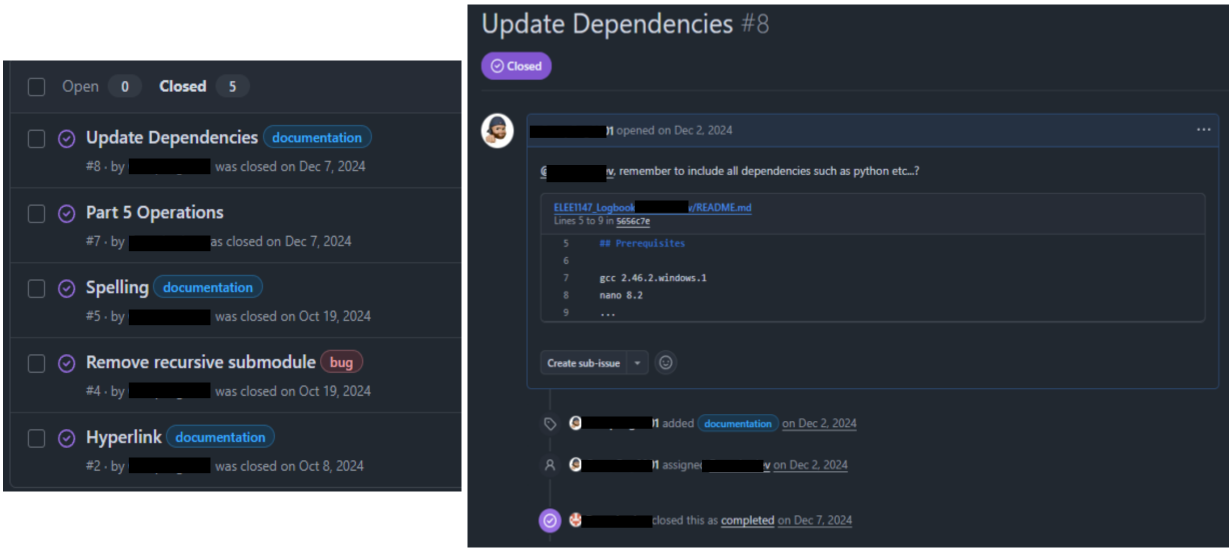

Figure 2: GitHub accessible formative and summative feedback (left: list of issues; right: detail of an issue)

To support readers unfamiliar with GitHub-based workflows, it is useful to clarify how feedback and feedforward mechanisms work in practice. A pull request presents proposed changes alongside the original work, enabling instructors or reviewers to comment on specific lines of code or text in a structured review interface. Issues function as threaded discussion spaces associated with the repository and are typically used to address higher-level conceptual, design or structural aspects of the work. Inline annotations appear directly adjacent to individual lines within a file, allowing for precise and contextualised feedback closely coupled to the student artefact. Branches represent parallel versions of the work created at a specific point in time, allowing students to experiment or revise independently without altering the original work until changes are reviewed and integrated via a pull request.

Evaluation

The adoption of GHC in HE brings with it a number of clear pedagogical and technological advantages, but also some notable challenges. From a constructivist perspective (Taber, 2017), the platform strongly supports learning through active engagement, as students construct understanding by iteratively producing, revising and reflecting on authentic artefacts rather than submitting static end products. The use of commits, branches, and pull requests situates learning within meaningful activity, aligning with principles of active learning (Coorey, 2016) and experiential education (Leih, Irawan, 2018).

A key strength of GHC is its alignment with industry practice, supporting authentic assessment and enhancing student employability through engagement with professional workflows. Within the substitution, augmentation, modification and redefinition (SAMR) model (Kurt, 2023), GHC extends assessment beyond substitution and into modification and redefinition, enabling forms of feedback, collaboration and engagement tracking that are difficult to achieve in conventional virtual learning environments. Commit histories provide transparent records of individual contributions, discouraging plagiarism and supporting fairer group assessment.

GHC's feedback mechanisms further reinforce its pedagogical value. Inline comments, issues and pull requests support dialogic, feedforward-oriented feedback that is embedded directly within the learning artefact, aligning with principles of effective formative assessment. Analysis of contribution patterns also enables early identification of disengagement, supporting timely instructional intervention. In large cohorts, automation tools such as GitHub Actions can deliver rapid formative feedback, allowing instructors to focus on higher-order conceptual guidance and reflecting emerging practices in digital pedagogy.

These benefits must be balanced against several limitations. The cognitive demands of Git-based workflows may be challenging for novices without prior technical experience. Institutional integration remains problematic, as GHC typically operates outside virtual learning environments, potentially fragmenting the learner experience, although plugins exist for some platforms (GitHub, 2025b). Issues of equity and access also arise, particularly for students with weaker digital skills or limited technical resources. Moreover, while automation may reduce some administrative overhead, instructors must still invest time in configuration and continuing technical support. Taken together, these advantages and disadvantages highlight both the promise and the complexity of deploying GHC in HE. Its affordances align strongly with the demands of modern digital industries, but its effective use requires careful pedagogical design, institutional support and attention to inclusivity.

Overall, GHC demonstrates strong pedagogical alignment with contemporary digital and industrial practices, but its effective use requires careful scaffolding, institutional support and sustained attention to inclusivity.

Conclusions

GHC represents a powerful bridge between academic study and industrial practice. Its strengths lie in authentic assessment, rich feedback opportunities and enhanced engagement tracking, as long as instructors act upon the provisos (mentioned above) for its effective deployment.

For practitioners, the following recommendations emerge:

- Introduce Git/GitHub gradually, with low-stakes formative tasks before graded assessments, like 'my first repository';
- Provide training resources (*e.g.* workshops, screencasts) to reduce the initial learning barrier;
- Combine automated feedback with human feedback to balance scalability and nuance;
- Explore integrations with institutional virtual learning environments to ensure smoother adoption.

Future directions might include: comparative studies of student attainment using GHC versus traditional submission systems; research into how GitHub's professional framing influences student motivation and identity formation.

Reference list

Chacon, S. and Straub, B. (2014) *Pro Git*. 2nd edn. New York: Apress. ISBN: 978-1484200773

Coorey, J. (2016) 'Active Learning Methods and Technology: Strategies for Design Education.' *The International Journal of Art & Design Education*, 35: 337-347. Available at: https://doi.org/10.1111/jade.12112 (Accessed: 01 October 2025).

Fowler, M. (2023) 'Software development in the age of AI.' *ThoughtWorks*. Available at: https://www.thoughtworks.com/insights/blog/software-development-age-ai (Accessed: 20 September 2025).

GitHub. (2023) *About GitHub*. Available at: https://github.com/about (Accessed: 20 September 2025).

GitHub. (2025a) *GitHub Classroom*. Available at: https://classroom.github.com/ (Accessed: 01 October 2025).

GitHub. (2025b) 'Connect a learning management system course to a classroom.' Available at: https://docs.github.com/en/education/manage-coursework-with-github-classroom/teach-with-github-classroom/connect-a-learning-management-system-course-to-a-classroom#about-connecting-an-lms-to-your-classroom (Accessed: 01 October 2025).

Kurt. S. (2023) 'SAMR Model: Substitution, Augmentation, Modification, and Redefinition.' Available at: https://educationaltechnology.net/samr-model-substitution-augmentation-modification-and-redefinition/ (Accessed: 19 December 2025).

Lieh, O.E. and Irawan, Y. (2018) 'Exploring Experiential Learning Model and Risk Management Process for an Undergraduate Software Architecture Course.' *2018 IEEE Frontiers in Education Conference (FIE)*, 1-9. Available at: https://doi.org/10.1109/FIE.2018.8659200 (Accessed: 01 October 2025).

Taber, K.S. (2017) 'The Role of New Educational Technology in Teaching and Learning: A Constructivist Perspective on Digital Learning.' In: Marcus-Quinn, A. and Hourigan, T. (eds.) *Handbook on Digital Learning for K-12 Schools*. Available at: https://doi.org/10.1007/978-3-319-33808-8_24 (Accessed: 19 December 2026).