Learning without Limits: Analysing the Usage of Generative AI in a Summative Assessment

Lee Clift
Department of Computer Science
Swansea University
Swansea, United Kingdom
l.a.clift@swansea.ac.uk

Olga Petrovska
Department of Computer Science
Swansea University
Swansea, United Kingdom
olga.petrovska@swansea.ac.uk

Abstract

This paper explores how Generative AI (GenAI) can be introduced within summative assessment components in software engineering education. We present an example of an assessment which allows learners to use GenAI in a freeform, constructionist manner, as part of a large, software development project. This work is inspired by previously executed AI-focused assessments and surveys, which explicitly indicate that learners on an Applied Software Engineering Degree Apprenticeship Programme want to formally learn how to use GenAI tools when programming and their employers want to see these skills from graduates. The learning outcome of the assignment was for learners to explore a typical developmental pipeline as a solo developer, moving from design to development to finished product. Learners were marked exclusively on their end product and understanding of application components, not the written code itself, resulting in an assessment where the end product and project were prioritised over foundational code (which was adequately assessed in other components). The results show that all learners used GenAI to some extent during their project, and in all cases, they found it beneficial for large programming tasks. Learners were generally able to produce a larger, more comprehensive and more ambitious project, compared to previous years. It is proposed that removing the barrier to GenAI - and demystifying it - can encourage a constructionist approach to its use, and normalise it as a potential tool for programming.

CCS Concepts

• **Applied computing** → **Education**; • **Social and professional topics** → **Software engineering education**; • **Computing methodologies** → *Artificial intelligence*.

Keywords

GenAI, software engineering, education, apprenticeship

ACM Reference Format:
Lee Clift and Olga Petrovska. 2025. Learning without Limits: Analysing the Usage of Generative AI in a Summative Assessment. In *Computing Education Practice (CEP '25), January 07, 2025, Durham, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3702212.3702214

CEP '25, January 07, 2025, Durham, United Kingdom

ACM ISBN 979-8-4007-1172-5/25/01
https://doi.org/10.1145/3702212.3702214

1 Introduction

Software Engineering is a consistently evolving field, where emerging technology is used within the industry to improve and optimise software development. Unfortunately, many learners enrolled in Higher Education degrees are not fully exposed to a realistic interpretation of this until they graduate and begin their careers, thus creating a gap between taught programmes and industry [9, 18].

As new technologies are introduced and integrated into industry and development cycles, it is the responsibility of educators to ensure that learners are kept up to date and that these technologies are integrated into their curriculum. By doing this, we ensure that – come graduation – learners will be prepared for the current technologies being used in the Software Engineering (SE) industry.

A key example of this new and emerging technology is Generative AI (GenAI), which is currently being integrated into Software Development lifecycles. The software engineering industry has been fast to adopt GenAI due to its ease of use within existing systems, and its time-saving possibilities [7, 21]. Unlike other technologies, GenAI has proven to be more controversial due to the risks around academic misconduct, collusion and plagiarism [1, 4].

We present a novel assessment, which combines the academic structure of a typical coursework assignment, with this emerging, academically controversial technology. This assignment was implemented to test where the inclusion of GenAI would result in more advanced student projects, as well as additional learning outcomes regarding the usage of GenAI tools in a simulated software engineering environment.

2 Background

The release of ChatGPT in 2022 prompted a growing interest in AI and its role in education [8, 23]. In 2023, there was a noticeable increase in publications related to GenAI, both in computing higher education and SE industry [22]. That year, UNESCO published their guidance on using GenAI in education and research [24].

After an initial period of uncertainty, educators are gradually embracing GenAI in their teaching practice; for instance, in chemistry [19] to improve critical thinking engagement, or in art and design [12] to enhance the creative process and prepare students to the realities of the AI-enabled world.

Computing education is no exception. CS educators integrate existing Conversational Agents (CAs) and develop customised GenAI tools to support student learning [2, 16, 20]. A notable example is the CS50.ai application, which incorporates a tool for explaining highlighted code, a tool for evaluating one's coding style, and a CS50 Duck chatbot, which answers questions related to the CS50 course [16]. CS50.ai was developed with so-called "pedagogical

guardrails" to guide students through the learning process rather than just give away the answers [16]. Hellas et al. analysed the use of LLM chatbot in 3 courses: Software Engineering with Large Language Models, Device-Agnostic Design, and Web Software Development, observing that its use differed considerably between the courses [10].

There are also other ways in which GenAI can be incorporated into CS and SE education that are not directly related to coding. For instance, Cámara et al. designed a formative assessment that specifically asks students to interact with GenAI when creating a UML diagram [6]. GenAI can also be used in user story design. Brockenbrough and Salinas conducted a study which required students to design user stories either completely independently or specifically using ChatGPT [3]. The study has shown that the students whose designs were assisted by GenAI produced comparatively higher-quality user stories [3].

While empirical studies suggest that GenAI can have a positive impact on learning, particularly for beginners, it is not as straightforward when it comes to academic integrity [22]. The issue of plagiarism is centuries old and numerous papers have been published on plagiarism in education. In the era of GenAI, defining what falls under plagiarism becomes a challenging task. As Hutson points out in [11], the paradigm is shifting and we are "reshaping our understanding of creativity, originality, and the collaborative writing process". This raises a couple of questions: 1) What constitutes "developing software" in this modern context? and 2) How can computing education address this new reality?

Mahon et al. attempt to shed some light in this respect; they look at different levels at which GenAI can be integrated into CS courses [17]. Our approach of an open-ended assignment, which allows the use of GenAI for ideation and automation of basic coding tasks, aligns with Level D as described by [17].

3 Methodology

A novel assessment was developed by the authors for a final year module, focused on Mobile Application Development, using the Dart programming language and Flutter API. Two new courseworks were developed, a short 2-week design task, including wireframes and a short design presentation, as well as a 7-week development task, culminating in a viva, following an established assessment pattern [14]. The initial design task was a typical academic assessment and followed the usual university regulations regarding prohibited tools and plagiarism/collusion. The second, longer task was more free-form. Learners were instructed to go about development in any way they wished, using any of the tools available to them, and given the long time frame, were encouraged to build ambitious, fully realised products. Learners had the freedom to design and develop an application about anything, whether academic or personal; which aided in motivation and application completeness [13]. The key characteristic of this coursework was that GenAI was allowed to be used, unlike the majority of assessments on the program. Learners were instructed beforehand that they may use it as they wish, and were given particular rules on how to cite generated code. At the point of submission, learners were also requested to fill out a self-reflection form, which asked them if, how, when and why they used GenAI in their project. Questions on this form asked about what GenAI model was used, when, why and how it was used, and to what extent they found it helpful over a number of metrics. This data was then used to inform future decisions on GenAI usage within the module, and the course.

In previous iterations of this module, learners were given coursework where they would have to critically analyse existing applications, and then create a new mobile application based on a pre-existing project specification. These assessments allowed access to typical open-book resources (such as online forums) and an IDE (Integrated Development Environment) [5] but prohibited generative technologies like GenAI, a practice which is well established within Software Engineering HE since it emerged. The motivation behind changing this is due to employers expressing how they wish to have learners with existing experience using GenAI tools, due to their belief that it is becoming a time-saving tool within the software development industry, alongside learners' own goals to expose themselves to these tools as well [7]. By allowing learners a chance to use this tool constructively, this fulfils the stated wishes of employers and learners, as well as enabling them to independently critically analyse how the tools work, and to what extent they feel they are useful. There were free to use the tool as much - or as little - as they wished, allowing them to organically decide if, when and how useful GenAI would be in a development situation.

In addition to the permitted tool allowance change, the time frame for development was extended. In previous years, students had been given a standard 2-week assessment deadline in alignment with the most modules on the programme, resulting in projects which were small in scale and rushed. Neither the timescale nor size of the project are typical of an industrial experience; therefore, these were changed for the new assessment, which was designed to allow substantially more time for development, namely seven weeks instead of the traditional two weeks. During this time, students were encouraged to use an incremental development cycle, slowly adding features as they were introduced in the course. The purpose of this was for learners to create substantially larger pieces of work, which would in turn allow them to learn more about the development process, especially since programming is typically best learnt practically and kinetically [15].

4 Results

Out of the 13 learners in the final-year cohort who took part in this assessment, 12 completed the post-assessment self-reflection survey. Results show that while 100% of the participants used some level of GenAI in the assessment, the tool they used, and the reason they used it varied. Over 50% of learners used ChatGPT, which is unsurprising, given the service's popularity [25]. Learners typically started to use GenAI at or before the midpoint of the assessment, 91.67% of learners using it by week 7 of the assignment. Further questioning led to learners showing what the main usage of GenAI was (Figure 2), with learners primarily using it for small modules of code, bug fixing and explanation of errors. This is further explored via textual responses, where 66% of students described how they used GenAI for generating small bits of code, i.e. widgets and tests, while 40% also wrote about its use for debugging logical and syntactical issues.

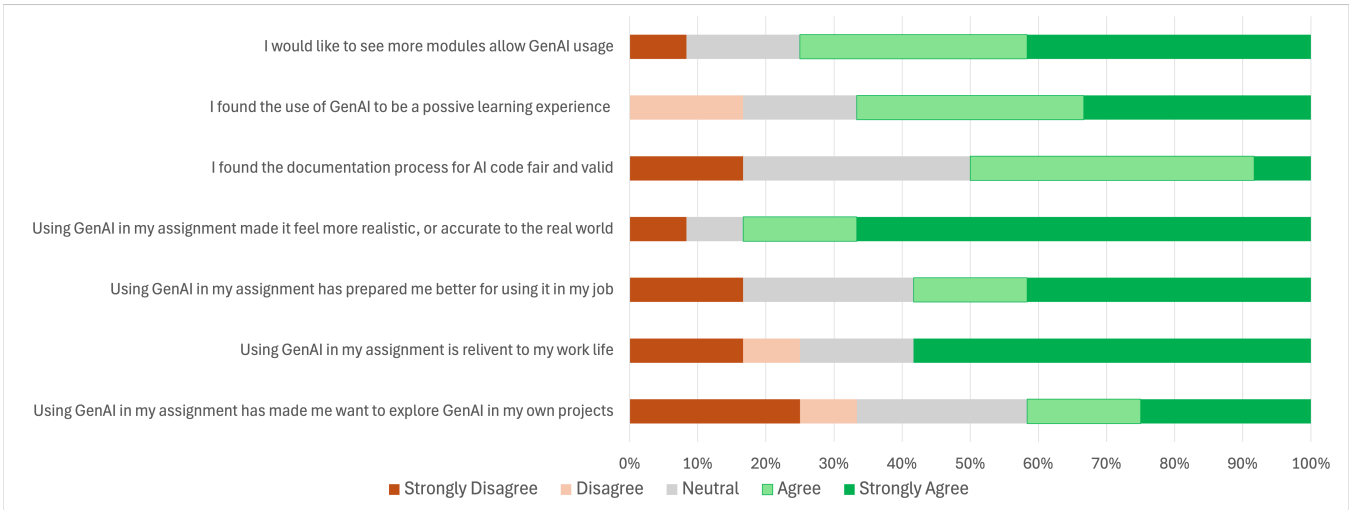

Figure 1: Selected 5-point Likert Scale questions, and the frequency of answers

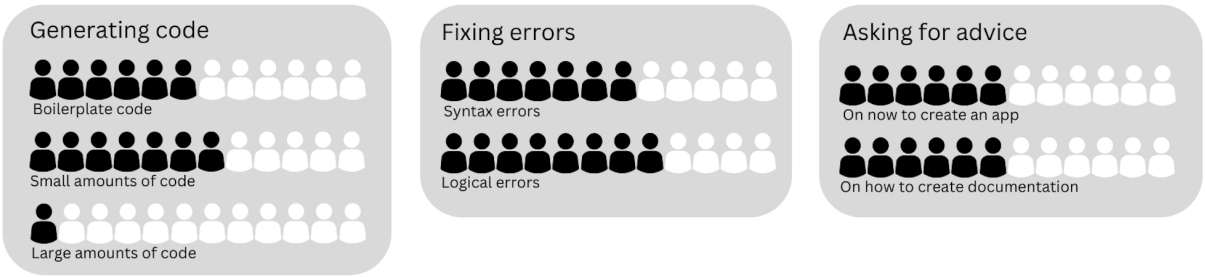

Figure 2: Learners' main purposes of using GenAI during development assignment

Learners were also asked to rate certain aspects on a 5-point Likert scale, with a variety of questions, ranging from how useful they found using GenAI on their assessments, to how frustrating they found citing their generated code. Figure 1 shows that the majority of learners felt either neutral or positive towards all aspects of the use of GenAI in their assessment. Over 80% of students felt that the inclusion of GenAI was a positive learning experience, and work-wise, made the assignment more realistic, and better-prepared learners for using GenAI in their work life.

Overall, students enjoyed the inclusion of GenAI tools in their assignment, mostly due to the potential future usage of GenAI. One learner stated, "*I feel like generative AI is one of the best learning tools available*". This positive sentiment was shared across all learners who completed the survey, with 100% stating they would use GenAI again and would like it to be continued to be allowed in future assessments.

5 Discussion

The results presented demonstrate a positive trend between learners being given freedom to explore GenAI tools in a summative assessment, and student enjoyment and project quality. Results show that a majority of students found the inclusion of AI a positive learning experience, wanting to see it included again. Learners also agreed that the use of GenAI would be a valuable experience for their workplace, and they would use it again, both in personal projects and at work. This reflects our hypothesis that including GenAI in a summative assessment would allow additional indirect learning in regard to the usage of GenAI.

Additionally, the projects created by learners were of very high quality, many of which, when 'viva'ed, were shown to be of publishable quality. The projects themselves were generally complete, with much of the progress being attributed to the access to GenAI and the increase in the programming speed associated with its usage. Much of the boilerplate or simple widgets of the apps could be generated quickly, allowing learners to focus their time on the more complex parts of the project.

However, the data does show two potential issues: learner perceptions of privacy and citing generated code. When asked about GenAI and privacy, many learners were not concerned - 83% of learners stated either they felt it wasn't an issue or held no strong feelings towards it. Similarly, when asked about citing generated code, 75% of learners found citing the generated code as either frustrating or neutral. Notably one learner went into detail on this point, stating *"Trying to record line by line what was me and what was ChatGPT was more effort that coding in the conventional way"*. This may indicate that while the inclusion of GenAI was a net positive,

how learners were requested to cite was negative. Therefore, while some aspects, such as the citing process may need to be reviewed for future iterations of the assessment, the inclusion of GenAI as a whole has been successful, and the results are encouraging for a rollout of future assessments in bigger cohorts.

Another potential limitation of the work is how learners had the potential to learn - and be assessed - on less core curriculum knowledge, due to the potential inclusion of GenAI, and the assistance it can provide. While this limitation wasn't formally tested, it likely occurred. To counteract this, other components of the module explicitly tested and encouraged this knowledge, specifically weekly practical labs, and an end-term in-class test.

The final limitation of this work is the notably small and constricted sample size. The cohort this assessment was given to consisted of only 13 students, with only 12 completing the follow-up self-reflective survey. While this is a clear constraint, the results are still helpful in indicating what can be done with GenAI in assessment. Replication of this assessment is encouraged in similar and different computing cohorts, to gather more conclusive results regarding the free usage of GenAI in assessments.

6 Conclusion

In conclusion, we presented the results of a summative programming assessment which included GenAI as a permitted tool. The results showed that learners enjoyed the experience, felt they learnt more than just the explicit syllabus, and produced projects which were larger in scale and better in quality than typical assignments.

In future, we plan to continue this work by integrating GenAI into more summative assessments, as the technology becomes more ubiquitous in industry, ensuring our assessment techniques stay in synchronisation with industrial realities. We also plan to investigate better ways of encouraging learners to document their generated code, as well as directly educate the social, ethical and security issues around software engineering and GenAI.

References

[1] Marc Alier, Francisco-José García-Peñalvo, and Jorge D. Camba. 2024. Generative Artificial Intelligence in Education: From Deceptive to Disruptive. *International Journal of Interactive Multimedia and Artificial Intelligence* 8, 5 (03/2024 2024), 5–14. https://doi.org/10.9781/ijimai.2024.02.011

[2] Patrick Bassner, Eduard Frankford, and Stephan Krusche. 2024. Iris: An AI-Driven Virtual Tutor for Computer Science Education. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (Milan, Italy) *(ITiCSE 2024)*. ACM, New York, NY, USA, 394–400. https://doi.org/10.1145/3649217.3653543

[3] Allan Brockenbrough and Dominic Salinas. 2024. Using Generative AI to Create User Stories in the Software Engineering Classroom. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*. 1–5. https://doi.org/10.1109/CSEET62301.2024.10662994

[4] Binglin Chen, Colleen M. Lewis, Matthew West, and Craig Zilles. 2024. Plagiarism in the Age of Generative AI: Cheating Method Change and Learning Loss in an Intro to CS Course. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (Atlanta, GA, USA) *(L@S '24)*. ACM, New York, NY, USA, 75–85. https://doi.org/10.1145/3657604.3662046

[5] Saraah Cooper, Ben Clinkscale, Briana Williams, and Myles Lewis. 2020. Exploring the Impact of Exposing CS Majors to Programming Concepts using IDE Programming vs. non-IDE Programming in the Classroom. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. ACM, New York, NY, USA, 1422. https://doi.org/10.1145/3328778.3372701

[6] Javier Cámara, Javier Troya, Julio Montes-Torres, and Francisco J. Jaime. 2024. Generative AI in the Software Modeling Classroom: An Experience Report with ChatGPT and UML. *IEEE Software* (2024), 1–10. https://doi.org/10.1109/MS.2024.3385309

[7] Christof Ebert and Panos Louridas. 2023. Generative AI for Software Practitioners. *IEEE Software* 40, 4 (2023), 30–38. https://doi.org/10.1109/MS.2023.3265877

[8] James Finnie-Ansley, Paul Denny, Andrew Luxton-Reilly, Eddie Antonio Santos, James Prather, and Brett A. Becker. 2023. My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises. *ACM International Conference Proceeding Series* (1 2023), 97–104. https://doi.org/10.1145/3576123.3576134

[9] Vahid Garousi, Gorkem Giray, Eray Tuzun, Cagatay Catal, and Michael Felderer. 2020. Closing the Gap Between Software Engineering Education and Industrial Needs. *IEEE Software* 37, 2 (2020), 68–77. https://doi.org/10.1109/MS.2018.2880823

[10] Arto Hellas, Juho Leinonen, and Leo Leppänen. 2024. Experiences from Integrating Large Language Model Chatbots into the Classroom. (2024). https://doi.org/10.48550/ARXIV.2406.04817

[11] James Hutson. 2024. Rethinking Plagiarism in the Era of Generative AI. *Journal of Intelligent Communication* 4, 1 (April 2024). https://doi.org/10.54963/jic.v4i1.220

[12] James Hutson and Bryan Robertson. 2023. Exploring the Educational Potential of AI Generative Art in 3D Design Fundamentals: A Case Study on Prompt Engineering and Creative Workflows. *Faculty Scholarship* 485 (2023). https://digitalcommons.lindenwood.edu/faculty-research-papers/485

[13] Amber Kemppainen, Gretchen Hein, and Nathan Manser. 2017. Does an open-ended design project increase creativity in engineering students?. In *2017 IEEE Frontiers in Education Conference (FIE)*. 1–5. https://doi.org/10.1109/FIE.2017.8190507

[14] John C. Knight and Thomas B. Horton. 2005. Evaluating a software engineering project course model based on studio presentations. In *Proceedings Frontiers in Education 35th Annual Conference*. S2H–21. https://doi.org/10.1109/FIE.2005.1612249

[15] Hong-Chan Ling, Kai-Lun Hsiao, and Wei-Chun Hsu. 2021. Can Students' Computer Programming Learning Motivation and Effectiveness Be Enhanced by Learning Python Language? A Multi-Group Analysis. *Frontiers in Psychology* 11 (2021), 600814. https://doi.org/10.3389/fpsyg.2020.600814

[16] Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. 2024. Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. ACM, New York, NY, USA, 750–756. https://doi.org/10.1145/3626252.3630938

[17] Joyce Mahon, Brian Mac Namee, and Brett A. Becker. 2024. Guidelines for the Evolving Role of Generative AI in Introductory Programming Based on Emerging Practice. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (Milan, Italy) *(ITiCSE 2024)*. ACM, New York, NY, USA, 10–16. https://doi.org/10.1145/3649217.3653602

[18] Damla Oguz and Kaya Oguz. 2019. Perspectives on the Gap Between the Software Industry and the Software Engineering Education. *IEEE Access* 7 (2019), 117527–117543. https://doi.org/10.1109/ACCESS.2019.2936660

[19] Harry E. Pence, Greta Hightower, Jackson Forlenza, Kaden Leonard, Alexandra McLellan, Amelie Suero, Bridget Amoah, Mariama Mbow, Sara Borner, Alexander Castillo, and Laura E. Pence. 2024. Using Generative AI Systems for Critical Thinking Engagement in an Advanced Chemistry Course: A Case Study. *Journal of Chemical Education* 101, 9 (09 2024), 3789–3794. https://doi.org/10.1021/acs.jchemed.4c00242

[20] Jaakko Rajala, Jenni Hukkanen, Maria Hartikainen, and Pia Niemelä. 2023. Call me Kiran – ChatGPT as a Tutoring Chatbot in a Computer Science Course". In *Proceedings of the 26th International Academic Mindtrek Conference* (Tampere, Finland) *(Mindtrek '23)*. ACM, New York, NY, USA, 83–94. https://doi.org/10.1145/3616961.3616974

[21] Asha Rajbhoj, Akanksha Somase, Piyush Kulkarni, and Vinay Kulkarni. 2024. Accelerating Software Development Using Generative AI: ChatGPT Case Study. In *Proceedings of the 17th Innovations in Software Engineering Conference* (Bangalore, India) *(ISEC '24)*. ACM, New York, NY, USA, Article 5, 11 pages. https://doi.org/10.1145/3641399.3641403

[22] Cigdem Sengul, Rumyana Neykova, and Giuseppe Destefanis. 2024. Software engineering education in the era of conversational AI: current trends and future directions. *Frontiers in Artificial Intelligence* 7 (Aug. 2024). https://doi.org/10.3389/frai.2024.1436350

[23] Ahmed Tlili, Boulus Shehata, Michael Agyemang Adarkwah, Aras Bozkurt, Daniel T. Hickey, Ronghuai Huang, and Brighter Agyemang. 2023. What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments* 10 (12 2023), 1–24. Issue 1. https://doi.org/10.1186/S40561-023-00237-X/FIGURES/13

[24] UNESCO. 2023. Guidance for generative AI in education and research.

[25] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Trans. Knowl. Discov. Data* 18, 6, Article 160 (apr 2024), 32 pages. https://doi.org/10.1145/3649506