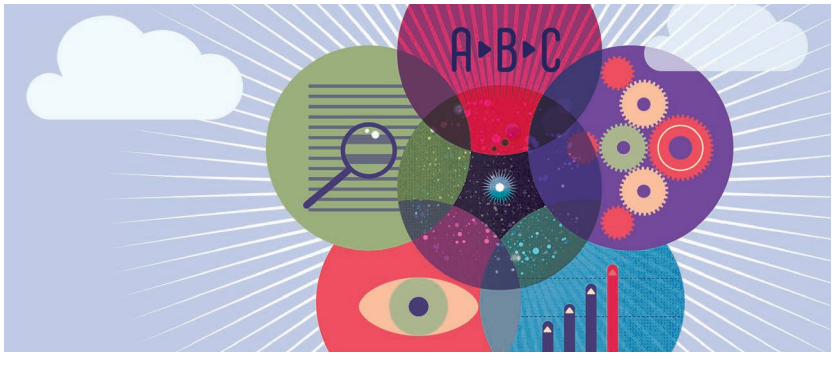
Generative AI in the Software Modeling Classroom

An Experience Report With ChatGPT and Unified Modeling Language

Javier Cámara, Javier Troya, Julio Montes-Torres, and Francisco J. Jaime, Universidad de Málaga

// *The use of generative AI chatbots in formative assessment can effectively gauge learning progress, increase the academic performance of students compared to a traditional methodology, and raise student awareness about the tradeoffs of employing generative AI in their work.* //



Digital Object Identifier 10.1109/MS.2024.3385309
Date of publication 8 April 2024; date of current version 10 October 2024.

ADMINISTERING FORMATIVE ASSESSMENTS to students soon after the information required to learn a given subject has been acquired can help to consolidate knowledge and lead to increased academic performance in summative assessments.[1,2] In the context of software engineering courses, formative assessments can involve administering tests in the form of classroom exercises or homework assignments (which can sometimes be graded and, hence, be partially used as summative assessments[3]).

The irruption of generative AI in the software engineering classroom, where students now employ tools such as Copilot (https://github.com/features/copilot) and ChatGPT (https://chat.openai.com), is radically changing the lay of the land,[4,5] rendering current assessment strategies obsolete due to the accuracy of generative AI at producing high-quality solutions, e.g., in programming tasks[6] or software verification.[7] Understandably, this is an area of utmost concern for software engineering instructors, who are starting to revise their assessment strategies in light of the latest developments.[8] Despite these efforts, little attention has been devoted so far to other areas that go beyond the scope of programming and that are also essential to the undergraduate software engineering curriculum, such as software modeling.[9]

One of the core skills taught in undergraduate software engineering courses concerns structural modeling using class diagrams in the Unified Modeling Language (UML),[10] which is the de facto standard notation employed to model object-oriented software systems. The traditional methodology for teaching structural modeling in UML typically involves face-to-face explanation of the underlying theoretical concepts, demonstrations of solutions for practical

exercises performed by instructors, and formative assessments that can be assigned as homework.

Until recently, these formative assessments could be used by instructors as an effective mechanism to gauge the degree to which concepts had been correctly assimilated by students. However, the widespread availability of generative AI tools casts a shadow of doubt over the effectiveness of these assessment methods and poses questions, such as the following: Can formative assessments still be relied upon as an effective method to improve learning and teaching efficiency in software modeling? Can software modeling instructors adapt their teaching methodology to incorporate generative AI and use it to their advantage?

Can formative assessments still be relied upon as an effective method to improve learning and teaching efficiency in software modeling?

Recent work on the assessment of generative AI for software modeling tasks[11] has shown that, while tools such as ChatGPT are able to generate UML software models (i.e., class diagrams) that are mostly syntactically correct, the quality of the results generated suffers from multiple shortcomings often related to the semantics of the entities represented in the diagram, which a human designer with a basic working knowledge of the notation should be able to spot. Inspired by that work, the hypothesis formulated in our study is that exposing students to interactive modeling tasks assisted by a generative AI tool (such as ChatGPT) can increase both their academic performance and their awareness about the tradeoffs of employing this technology as well as enable instructors to effectively gauge required areas of improvement.

To contribute toward adapting software modeling teaching methodologies to the new status quo, our study tests this hypothesis by exploring the following research questions:

- *Q1*: Can instructors detect required areas of improvement for students by gauging their perception about the quality of software models obtained via generative AI?
- *Q2*: Do students who have carried out formative assessments with the support of generative AI perform better in software modeling summative assessments compared to students who did not employ such tool support?
- *Q3*: What are the perceived tradeoffs of students with respect to the use of generative AI for software modeling tasks (in terms of, e.g., effort or quality of the solution generated)?

Methods

This section provides an overview of the methods employed in our study, starting with course teaching and assessment methods, followed by data collection and analysis.

Teaching and Assessment Methods in the "Introduction to Software Engineering" Course

The study was conducted during the academic year 2022–2023 in the context of "Introduction to Software Engineering," which is a compulsory course that students take during the second semester of their second year of the software engineering and computer science degrees at the School of Computer Science in the University of Malaga. The course is taught over 15 weeks and is divided into a theoretical and a practical component. The course comprises 18 onsite lectures in the theory classroom as well as 12 lab sessions. The practical component incorporates a learning-by-doing component[12] in which students have to develop a small software project collaboratively with six to eight of their peers. UML class diagram modeling is taught and assessed within the theory component of the course, independent of the practical component.

Teaching Methodology. To conduct our study, a student control group was taught UML class diagram modeling following a traditional methodology, and an experimental group was taught with a modified methodology that included the support of generative AI. Out of an overall population of 242 students, the experimental group was formed by 72 students who volunteered, incentivized by a small increment in their course score (compare the details in the "Threats to Validity" section).

- *Traditional methodology*: Software modeling was taught

during two lectures (each of 110 min in duration) in the classroom, where instructors 1) performed a face-to-face explanation of the underlying theoretical concepts; 2) illustrated these concepts with examples; and 3) assigned three practical exercises to students, who had 10–15 min to try to solve each exercise before they were collaboratively solved on the whiteboard with the participation of the entire class, guided by the instructor. Classroom lectures were complemented by a lab session during which the instructor illustrated how to model one of the examples seen in the classroom using the modeling package Visual Paradigm (https://www.visual-paradigm.com/).

- *Methodology supported by generative AI*: Software modeling was taught following the same procedure employed by the traditional methodology but including an additional formative assessment in which students had to use generative AI—ChatGPT in particular—to carry out two modeling tasks with the support of the chatbot. (A repository including artifacts, such as student handouts, surveys, and anonymized results, can be found at https://github.com/javier-camara/teaching-mod-UML-ChatGPT.)

In the handout of the exercise, students received instructions for the use of ChatGPT, including how to write prompts to obtain class diagrams as a response and a sample conversation. The explanations focused on the PlantUML notation (http://plantuml.com) because it includes a textual notation that can be automatically translated into a graphical representation. Next, the two tasks were described.

In the first task, a set of requirements about a management application for theaters and plays was provided. Students were asked to obtain a class diagram for the application by entering a prompt. If they were not satisfied by ChatGPT's response, they could chat with it to improve the response. If the best response obtained by ChatGPT after several interactions did not satisfy the student, they could improve it manually.

In the second task, students were given a class diagram about books, copies, and authorship consisting of four classes. They were asked to try to obtain the same class diagram by entering a prompt into ChatGPT. If a similar diagram was not obtained, they could continue the conversation with ChatGPT to try to obtain a diagram as close as possible to the original.

Student Assessment. The theoretical and practical components of the course had a weight of 20% and 80% of the final score, respectively. The summative assessment of the theoretical component was divided into a multiple-choice test and a UML class diagram modeling exercise, which had weights of 75% and 25%, respectively. In the UML exercise, students received the description of a system for which they were asked to write a class diagram. During the exercise, students did not have any access to the Internet or to class materials. The score of the UML exercise ranged from zero to 10. For marking, instructors initially assigned the highest score to the student and started subtracting points based on the errors found in diagrams: the absence of classes, superfluous classes, incorrect multiplicities, incorrect inheritance relations, etc. All instructors agreed upon the subtraction mechanism, consulting with each other cases that were not entirely clear.

Our study focuses exclusively on the UML class diagram modeling exercise as the part of the summative assessment used to measure the academic performance of all students (both in the control and experimental groups).

Data Collected

To answer the research questions in our study, we collected information from multiple sources.

Formative Assessment (AI-Supported) Submissions. For the two exercises described in the methodology supported by generative AI, students had to submit their conversation with ChatGPT, the final diagram obtained, and a description of the problems encountered in the solution returned by ChatGPT and of what aspects had to be improved to obtain a better solution. Students had three weeks to complete their assignment. Overall, 72 submissions were collected.

Formative assessments were evaluated focusing on the skills students should have acquired when learning to model class diagrams. Concretely, exercises were assessed for consistency, i.e., checking whether students made a genuine attempt, taking into consideration the degree to which they were able to spot errors in ChatGPT's responses related to 1) entities, 2) attributes, 3) inheritance, 4) associations, 5) redundant associations, and 6) multiplicities. To have a reference about the accuracy of student perceptions, instructors independently evaluated ChatGPT's

results obtained by every student during the formative assessment.

Survey. At the end of the submission of their formative assessment with ChatGPT, students were asked to fill in a survey with three objectives: 1) assessing whether students were able to identify the strengths and weaknesses of ChatGPT based on the modeling task they had been asked to perform (with results feeding into Q1 and Q3), 2) fostering critical thinking abilities in students by asking about perceived tradeoffs regarding the use of generative AI technology for modeling (with results feeding into Q3), and 3) collecting contextual information about the experience of the students that might be relevant but was not directly captured in other questions. The survey comprises 24 questions, out of which the first block (15 questions) target objective 1 and are based on Likert scales that include the categories none, almost none, sometimes, almost all, and all for questions such as, "Do diagrams include all the necessary relations?" The second block (six questions) targets objective 2 and combines yes/no questions, such as "Does ChatGPT always generate the same result for the same prompt?" with open-ended questions, such as "Do you think that ChatGPT is a suitable tool for generating useful models?" Finally, the third block contains three open-ended questions so that students can provide further context and thoughts about their experience. A total of 55 answers were collected.

Summative Assessment Grades. The results of the UML class diagram modeling summative assessment described in the "Student Assessment" section were also collected to feed into the analysis required to answer Q2. Overall, the results for 242 students were collected.

> In the handout of the exercise, students received instructions for the use of ChatGPT, including how to write prompts to obtain class diagrams as a response and a sample conversation.

Statistical Techniques

Q1: Gauging Learning Progress. This question is only applicable to students who took the AI-supported formative assessment. To determine the accuracy of student perception with respect to the quality of the class diagrams generated by ChatGPT, we conducted a Pearson correlation test between survey answers (the first block of 15 questions) provided by students and those provided by the independent evaluation of the instructors, who answered the same survey questions based on the ChatGPT conversation logs and results generated by students during the formative assessment.

Q2: Student Performance. Of the 242 students involved in the study, 72 (≈30%) belong to the group in which the methodology supported by generative AI was implemented (the ChatGPT group). The remaining 170 followed the traditional methodology (the control group). To determine whether the ChatGPT group performed better than the control group in the UML exercise, statistical hypothesis testing was employed.

The first step was a normality assessment by the Shapiro–Wilk test.[13] This test evaluates the null hypothesis (H_0) that the sample comes from a normally distributed population. Hence, the alternative hypothesis (H_a) states that the population from which the sample comes is nonnormal.

Once the normality tests were performed, the results revealed that the comparison of the arithmetic means of the ChatGPT and control groups should be made with a nonparametric test. The Wilcoxon rank sum test[14] was chosen for this purpose, as it is the most widely used nonparametric test for comparing two independent populations. A one-sided test was run considering the following null and alternative hypotheses: H_0 = the arithmetic mean of the UML exercise marks is lesser for the ChatGPT group than it is for the control group; H_a = the arithmetic mean of the UML exercise marks is greater for the ChatGPT group than it is for the control group.

Every statistical test was performed at a significance level of $a = 0.05$ thus, p values were obtained with a 95% confidence interval. A p value of $>a$ suggests that there is evidence to accept H_0. On the contrary, if the p value is less than a, H_0 is rejected, and H_a is assumed to be true.

Results

Survey

Figure 1 summarizes the results of the first part of the survey, which relates to the quality of the diagrams generated by ChatGPT (as identified both by students and instructors) and is divided into three blocks. Figure 1(a) corresponds to the inclusion of necessary elements in the diagram, showing that students identify the presence of elements in the categories "all of them" or "almost all" in more than 90% for classes, more

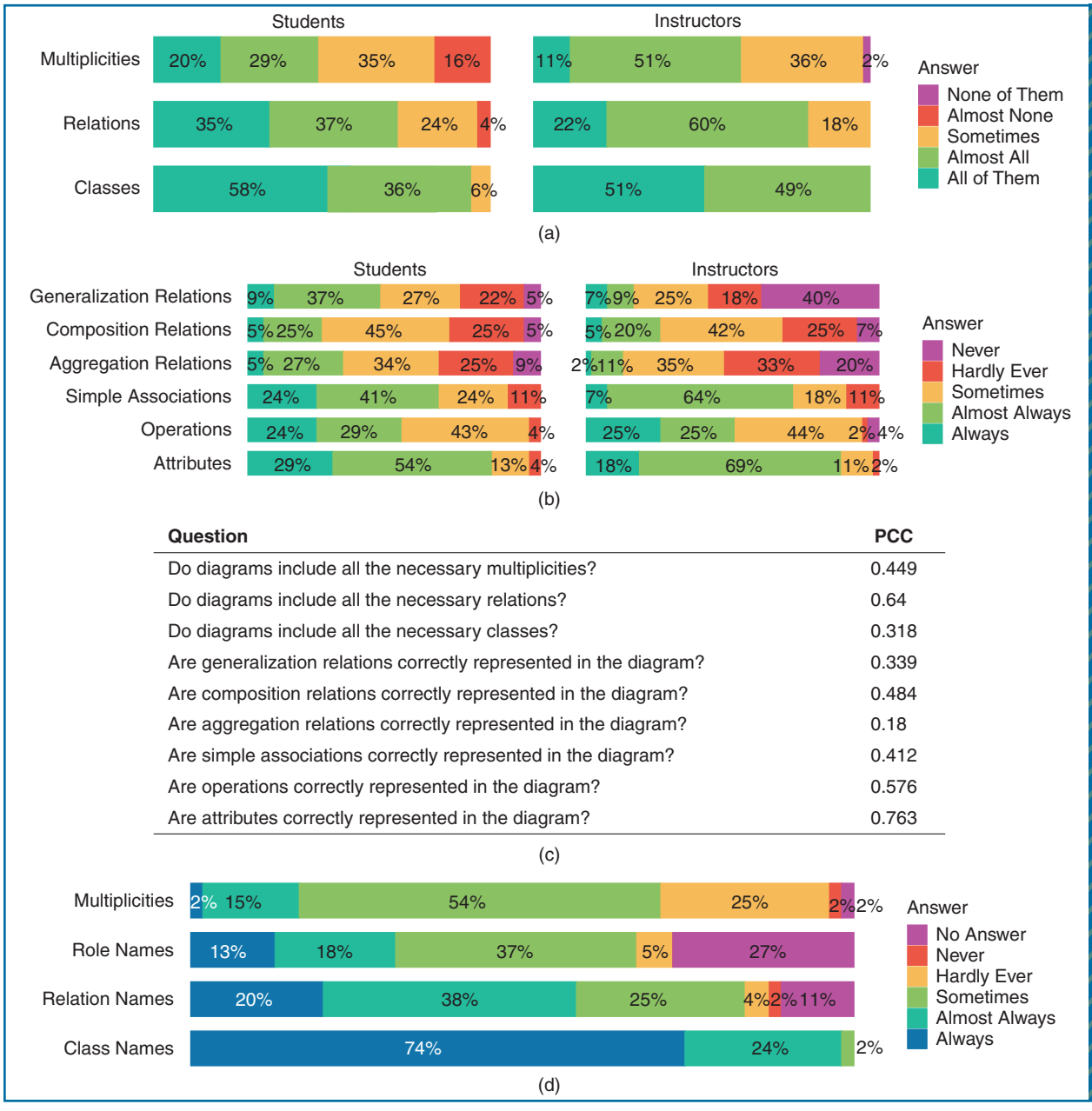


FIGURE 1. Results about the quality of the diagrams generated by ChatGPT identified by students. (a) Inclusion: do the diagrams include all of the necessary elements? (b) Syntactic correctness: are elements correctly represented in the diagram? (c) Pearson correlation coefficient (PCC) between student and instructor answers. (d) Semantic correctness: are element labels and values meaningful/correct?

than 70% for relations, and less than 50% for multiplicities.

Figure 1(b) concerns the syntactical correctness of the elements represented and shows that attributes, simple associations, and generalization relations are the categories with the highest percentages in the categories "always" and "almost always" (≈83%, 65%, and 46%, respectively).

Figure 1(c) includes the Pearson correlation coefficients between student and instructor answers, which tell us how accurate the perceptions of students in Figure 1(a) and (b) were. We can observe that, although students did reasonably well (all correlation coefficients are positive), they were better at identifying missing relations (0.64) compared to missing classes (0.318), while they struggled to identify whether aggregation (0.18) and generalization (0.339) relations were correctly represented.

Figure 1(d) relates to the semantic correctness of elements and shows that, while class names are found to be meaningful (almost) always (98%), on the other end of the spectrum, multiplicities and role names score 17% and 31%, respectively.

Figure 2 shows results about the experiences of students with the tool. Figure 2(a) shows that most students agree that ChatGPT is an interesting tool to learn class diagram modeling (≈71%), while it also has to improve to provide better diagrams faster (≈69%). Figure 2(b) shows that when the same prompt is supplied more than once to ChatGPT, relations and multiplicities are the elements that experience most variability—99% of students acknowledge that ChatGPT does not generate results deterministically. Finally, Figure 2(c) shows that almost 60% of students required five or more attempts to generate the desired result.

Summative Assessment Performance

Figure 3 shows a boxplot for the UML summative assessment scores in the control and experimental groups. The arithmetic means of the scores of the ChatGPT and the control groups are, respectively, 6.056 and 4.62. Running the Shapiro–Wilk test for each sample produced a p value of 9.878×10^{-5} for the ChatGPT group and a p value of 0.001 for the control group. In both cases, we reject H_0 and conclude that every sample comes from a nonnormal population. Student's t test requires the assumption of normality to be met. Therefore, the Wilcoxon rank sum test was chosen

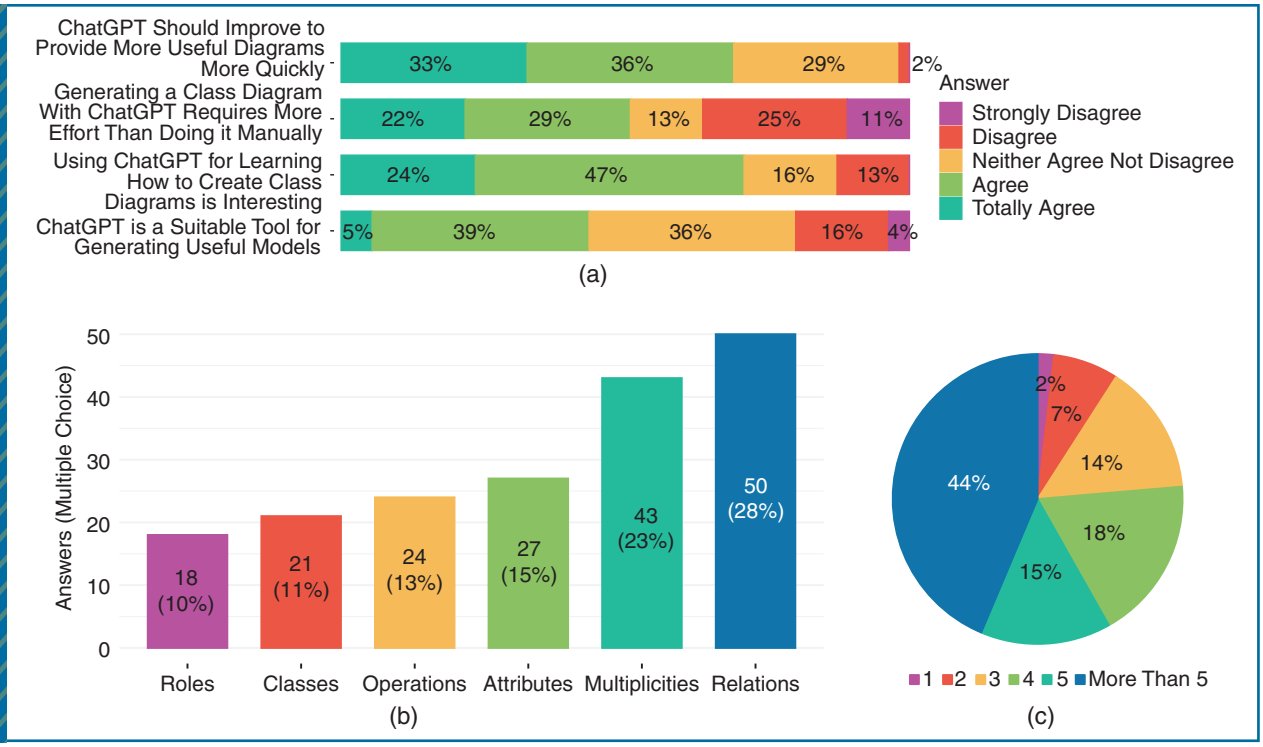


FIGURE 2. Survey results about student experience with the use of ChatGPT for class diagram modeling. (a) Student opinion about the tool suitability and tradeoffs for the modeling tasks. (b) Varying class diagram elements for the same prompt. (c) Attempts required.

to compare the means of the two populations. The p value for the one-sided test was 6.335×10^{-5}, which clearly leads to the conclusion that H_a has to be true. Consequently, the ChatGPT group performed clearly better than the control group in the UML summative assessment.

Discussion

Q1: Learning Progress Gauging

The results in Figure 1 and described in the "Results" section show that students did not have entirely accurate perceptions of the performance of ChatGPT in class diagram modeling. Although they did a fair job at detecting some strengths and weaknesses of ChatGPT that are in line with the results obtained by a recent work on the assessment of generative AI for software modeling tasks,[11] they struggled at correctly identifying the lack of inclusion of some elements, like classes, or the correct representation of some elements, notably, aggregation relations and generalizations. These results indicate that it is feasible for instructors to gauge with a relatively high level of accuracy which areas students still need to improve in, providing a valuable tool to inform potential changes in their teaching strategy.

Q2: Student Performance

Students who followed the methodology supported by generative AI received the same training as students who followed the traditional methodology. Additionally, they performed the two tasks where they had to use ChatGPT. As shown in the data of Figure 3, we have evidence supporting that, in general, students who participated in the ChatGPT formative assessment performed better in the summative assessment about UML class diagram modeling.

Our interpretation of these results is that students who participated in the ChatGPT formative assessment employed more time reasoning about the different elements involved in UML class diagrams, such as attributes, classes, and associations. Deciding whether ChatGPT was good at modeling these elements required them to think about the kind of solution that should be returned by ChatGPT, so they needed to look carefully into how to properly model class diagrams.

Q3: Perceived Tradeoffs

Survey results show that, while students in general have a positive opinion about the usefulness of ChatGPT to learn class diagram modeling, half of them also acknowledge that employing the tool to obtain a fully fledged solution requires more effort than building diagrams manually from scratch. Moreover, many students are not convinced about the usefulness of the diagrams generated. Interestingly, these results are aligned with reflections provided by many students in the comments section of the survey, who explicitly acknowledge that ChatGPT does not seem to understand the problem at hand and that a thorough revision of the results is always required. However, this does not seem to be considered a problem, with multiple students indicating that the tool is useful to obtain a preliminary low-cost solution that can be later improved upon manually. These reflections are indicators that the main tradeoffs perceived by students are that little effort is required to obtain low-quality solutions, whereas correct solutions require much more effort—even more than the effort required by manual correction.

Threats to Validity

The data from the control and experimental groups were obtained from students who were trained by different instructors. Despite this difference, all instructors followed the same teaching and assessment methods

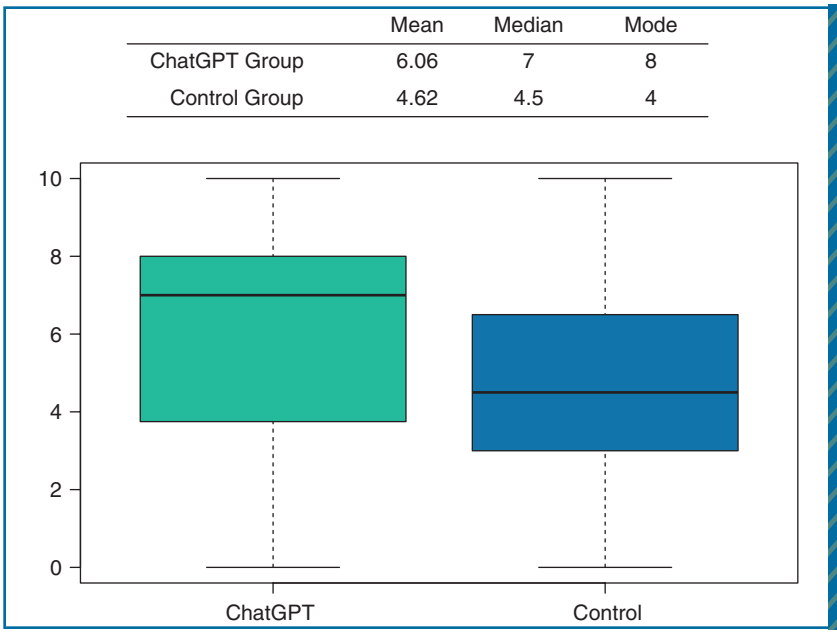
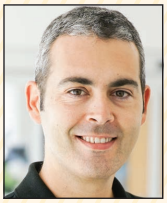


FIGURE 3. Comparison of the control and experimental group performances in the summative assessment.

ABOUT THE AUTHORS



JAVIER CÁMARA is an associate professor of computer science at the University of Malaga, 29071 Malaga, Spain, and honorary visiting fellow at the Department of Computer Science, University of York, YO1 5GH Heslington, U.K. His research interests include self-adaptive and autonomous systems, software architecture, formal methods, and AI-enabled systems. Cámara received his Ph.D. degree with honors in computer science from the University of Malaga. Contact him at https://javier-camara.github.io/ or jcamara@uma.es.

JAVIER TROYA is an associate professor of software engineering at the University of Malaga, 29071 Malaga, Spain. His research interests include model transformation testing, uncertainty modeling, and digital twins. Troya received his Ph.D. degree with honors in computer science from the University of Malaga. Contact him at https://javiertroyauma.github.io/ or jtroya@uma.es.

JULIO MONTES-TORRES is an interim professor at the University of Malaga, 29071 Malaga, Spain. Montes-Torres received his M.S. degree in computer science from the University of Malaga. His research interests include the application of generative and discriminative machine learning models to molecular data-based prognosis and other classification problems. Contact him at julio@lcc.uma.es.

FRANCISCO J. JAIME is an interim professor at the University of Malaga, 29071 Malaga, Spain. Jaime obtained his Ph.D. degree with honors from the University of Malaga in computer arithmetic and application-specified integrated circuit implementation. Contact him at franj@uma.es.

and synchronized often to guarantee homogeneous practices. Moreover, the ChatGPT formative assessment was taken by 72 of the 242 students, meaning that the split between the experimental and control groups is approximately 30%/70%. This is due to the fact that imposing up front which students would participate in the formative assessment was deemed unfair because the fraction of students (unwillingly) left out in the control group would not be able to reap the benefits of the approach. Hence, students were incentivized with an extra +0.5/10 in their overall course score to volunteer. In any case, the 30%/70% split is not considered to be a problem because the size of the experimental group is enough to deem as reliable the p values obtained in nonparametric tests like the ones employed, which are considered to be more robust than parametric tests, even with small samples.

An additional threat to validity concerns the potential imbalance in learning experiences between the traditional methodology and generative AI groups. It is important to clarify that both groups, including those following the traditional methodology, were assigned a series of modeling homework exercises without a time limit. This commonality partially mitigates concerns regarding time allocation discrepancies. However, we acknowledge that the additional exposure to generative AI tasks may have conferred an advantage to the experimental group. To address this in future studies, we will consider equalizing the extent of additional experiences between groups, ensuring a more balanced comparison of learning outcomes.

Moreover, our study acknowledges a limitation due to not conducting initial assessments of students' software modeling knowledge. This omission means we could not account for baseline knowledge differences between groups. However, this concern is mitigated, as our participants are second-year students with very limited or no exposure to software modeling, both academically and professionally. Therefore, the likelihood of significant discrepancies in software modeling experience between the groups is minimal. We also acknowledge the possibility of varying motivation levels and inherent abilities among students based on group assignment. However, this concern is mitigated by the fact that the population includes students in both the control and the experimental group from every cohort that participated in the study and the fact that all cohorts have similar entry-level academic requirements.

The irruption of generative AI poses new challenges to assessment strategies in

software engineering, particularly when it comes to take-home assignments, in which instructors are unable to directly supervise student work. However, our study has shown how actively fostering the use of generative AI tools in the software modeling classroom has the potential to increase the academic performance of students, to make them aware of the tradeoffs of using such technology, and to preserve the ability of instructors to effectively gauge learning progress despite the use of generative AI by students.

Although our study has been carried out in the specific context of using ChatGPT for UML class diagram modeling, we posit that the benefits of using our approach are generalizable to other settings in the software engineering classroom. Implementing analogous strategies requires careful assessment design that enables students to 1) explore in a directed way the use of available tools and 2) critically reflect upon their experience so that they can make better informed future decisions about the use of AI technology in a given context of the software engineering process. Such assessment design can be supported and informed by prior knowledge about the behavior and tradeoffs of the technology employed, obtained, for instance, from exploratory studies like the one used in our work.[11]

We believe that devising teaching approaches that share the philosophy embedded in our study can help software engineering instructors train competent professionals able to assess the suitability of using generative AI technology in different contexts.

Acknowledgments

This work was partially funded by the Spanish Government (Fondo Europeo de Desarrollo Regional/Ministerio de Ciencia e Innovación–Agencia Estatal de Investigación) under Projects TED2021-130523B-I00 and PID2021-125527NB-I00.

References

1. J. D. Karpicke and H. L. Roediger III, "Expanding retrieval practice promotes short-term retention, but equally spaced retrieval enhances long-term retention," *J. Exp. Psychol., Learn., Memory, Cognition*, vol. 33, no. 4, 2007, Art. no. 704, doi: 10.1037/0278-7393.33.4.704.
2. H. L. Roediger III and J. D. Karpicke, "The power of testing memory: Basic research and implications for educational practice," *Perspectives Psychol. Sci.*, vol. 1, no. 3, pp. 181–210, 2006, doi: 10.1111/j.1745-6916.2006.00012.x.
3. J. Camara and D. Garlan, "Learning by redoing: An experimental study on the impact of repetition of formative assessments in a formal methods course for software engineers," *IEEE Softw.*, vol. 40, no. 6, pp. 95–101, Jul. 2023, doi: 10.1109/MS.2023.3291400.
4. M. Halaweh, "ChatGPT in education: Strategies for responsible implementation," *Contemporary Educ. Technol.*, vol. 15, no. 2, 2023, Art. no. ep421, doi: 10.30935/cedtech/13036.
5. M. Montenegro-Rueda, J. Fernández-Cerero, J. María Fernández-Batanero, and E. López-Meneses, "Impact of the implementation of ChatGPT in education: A systematic review," *Computer*, vol. 12, no. 8, 2023, Art. no. 153, doi: 10.3390/computers12080153.
6. E. L. Ouh, B. K. S. Gan, K. J. Shim, and S. Wlodkowski, "ChatGPT, can you generate solutions for my coding exercises? An evaluation on its effectiveness in an undergraduate Java programming course," 2023, *arXiv:2305.13680*.
7. S. Jalil, S. Rafi, T. D. LaToza, K. Moran, and W. Lam, "ChatGPT and software testing education: Promises & perils," in *Proc. IEEE Int. Conf. Softw. Testing, Verification Validation Workshops (ICSTW)*, 2023, pp. 4130–4137, doi: 10.1109/ICSTW58534.2023.00078.
8. T. Phung et al., "Generative AI for programming education: Benchmarking ChatGPT, GPT-4, and human tutors," *Int. J. Manage.*, vol. 21, no. 2, 2023, Art. no. 100790.
9. M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern, and W. Visser, "SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering," *Computer*, vol. 48, no. 11, pp. 106–109, Nov. 2015, doi: 10.1109/MC.2015.345.
10. M. Gogolla, *Unified Modeling Language*. Boston, MA, USA: Springer US, 2009, pp. 3232–3239.
11. J. Cámara, J. Troya, L. Burgueño, and A. Vallecillo, "On the assessment of generative AI in modeling tasks: An experience report with ChatGPT and UML," *Softw. Syst. Model.*, vol. 22, no. 3, pp. 781–793, 2023, doi: 10.1007/s10270-023-01105-5.
12. M. Bernreuther and H.-J. Bungartz, "Learning by doing: Software projects in CSE education," in *Proc. Int. Conf. Comput. Sci.*, V. N. Alexandrov, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, Eds., Berlin, Germany: Springer Berlin Heidelberg, 2006, pp. 161–168, doi: 10.1007/11758525_22.
13. S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, nos. 3–4, pp. 591–611, 1965, doi: 10.2307/2333709.
14. M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, 2013.