# Value-Added Grading of AI-Assisted Papers

Eric Braude(✉)

Boston University, Boston, MA 02215, USA
ebraude@bu.edu

**Abstract.** The advent of AI generators such as chatGPT is significantly affecting the workplace as well as teaching and learning in higher education. This includes education in systems analysis, design, and programming. Working IT professionals will routinely rely on AI generation, we believe. Consequently, students should learn to leverage AI tools. We describe a way to prepare systems analysts and developers for a professional environment in which AI generation is ubiquitous and we explain an approach to evaluating graduate work prepared with the aid of AI generation. It is based on the *value added* in students' work: their chosen prompt sequences and their edits, deletions, and additions to AI-generated material. Although professionals are judged by their products, we believe that value-added will be a key professional goal in the AI-rich environments of the future because it distinguishes the end-products of professionals who better leverage AI.

**Keywords:** Assessment · IT education · Grading · Prompt engineering · Systems analysis and design · AI generation · Software development

## 1 Introduction

After much anticipation and discussion (e.g., Floridi and Chiriatti [2]), chatGPT was released in November 2022 and is affecting the workplace, as well as teaching and learning in higher education. This includes education in systems analysis, design, and development. Professionals are judged by their products, but we believe that value-added will be a key professional goal in the AI-rich environments of the future because it distinguishes the quality of the end-products of those who better leverage AI. Software systems professionals will routinely rely on AI generators, and students should learn to fully utilize these tools.

The use of tools in higher education is frequently encouraged, and we typically assess student work by inspecting the end-product. AI generation, however, is powerful enough to do at least some of the significant work we traditionally expect from students. For example, even the following prompt to chatCPT generates a complete, albeit simple, game that "will help you grasp the concept" of a full game: *Give the Python code for a console-based videogame to save the Earth from environmental collapse*. Additional prompts can make the game increasingly challenging.

Although some instructors claim to easily detect AI-generated material, and although tools exist that claim to detect it, such tools are not sufficiently reliable (Sadasivan


T. Zlateva and G. Tuparov (Eds.): CSECS 2023, LNICST 514, pp. 399–406, 2023.
https://doi.org/10.1007/978-3-031-44668-9_31

et al. [8]) and we can't fairly convict students of plagiarism without proof. Neither suspicion nor tools supply such proof. In any case, we believe that the use of AI should be encouraged and rewarded, not punished. Approaches to assignments and evaluation must adapt.

The unprecedented[1] nature and dynamic of AI generation is such that very early study of its effects on education, and corresponding practical adjustments, despite being necessarily incomplete, are called for without delay: hence this report and work-in-progress.

We began this work with a course, "Systems Analysis and Design," which started in the second week of March 2023, four months after the introduction of chatGPT. We continued it in a course on Advanced Java Programming, beginning in May.

We identified the following research questions.

1. How do we educate software systems professionals for an environment in which AI generation is ubiquitous?
2. How can we avoid passive, over-reliance on AI generation, where students do not sufficiently absorb the required concepts?
3. Is there a way to effectively evaluate the systems analysis or programming knowledge of graduate students when they use AI?

Our response to the first question was to incentivize students to leverage AI generation to the full throughout the course and to provide guidance on prompts, as exemplified in Sect. 1.1 below. Our response to the second and third research questions was to evaluate student work based on their *value added*—their own prompt sequences and their improvements to the resulting chatGPT end-products.

Students were permitted to opt out of using AI generation upon request, but they were required to justify this.

### 1.1 Prompt Guidance

Prompt advice—now ubiquitous on the Web—was shared throughout the term, specialized to systems analysis and programming. The following were typical.

*GENERIC PROMPT ADVICE*[2]*:*
*Be specific about word count and make it higher than you will eventually need. Understand the chatbot's limitations. Use clear and concise prompts. Ask the AI to add more when you need more. Ask the AI to reformulate its response if you need it. Use key words. Keep it simple. Review AI output for accuracy. For code generation, request pre- and postconditions; request tests.*

*METAPROMPT EXAMPLES*

- *I want to create a role-playing video game and I need its overall states and transitions. Give me some chatGPT prompts to get the best results.*

[1] Hundreds of the world's most prominent AI scientists signed a statement in May 2023 that "Mitigating the risk of extinction from AI should be a global priority alongside other societal-scale risks such as pandemics and nuclear war." (https://www.safe.ai/statement-on-ai-risk).

[2] Edited from https://www.elevatodigital.com/10-tips-and-tricks-for-using-chatgpt/.

- *What are some helpful prompts that I might ask you (chat GPT) about project risks?*

*PROJECT IDEA EXAMPLE*
*Give a numbered list of 8 to10 requirements for a role-playing video game about conservation, resident on a PC. Use "shall." Be precise.*

*CODE IMPROVEMENT EXAMPLE*
*Reorganize the following code to make it more modular, more readable, and with less nesting. Leave the comments intact.*

*CODE GENERATION TEMPLATE EXAMPLE*[3]
*I want you to act as a grandmaster Java software developer who has extensive experience writing Java code for mission-critical Java applications and is capable of teaching Java to other advanced Java users. I will give you a specification of different aspects of a Java application, and I want you to reply with Java code using one code block per Java class. I want you to reply only with the Java code and nothing else. Do not write explanations. I want you to provide an "Intent" comment just above each Java class that explains the intent of each class. Use no more than two sentences for each of these comments. This comment must not include any implementation details and must not mention the name of any Java classes. The following is a bad example which includes the name of a class: "The intent of this class is to average ten numbers using the NumbersList Class". Do not follow that example. Instead, say: "The intent of this class is to average numbers". Exclude implementation details, do not mention what classes the implementation will use, and include only the intent of the class. For each method that is not a getter or setter, I want you to provide preconditions and postconditions. A precondition is something that must be true before the method executes. A postcondition is something that must be true after the method has been completed.*

We continue to work on testing and improving our advice (e.g., is anything lost by prompting "Provide …" instead of "I want you to provide …"?).

## 2 Related Work

Finnie-Ansley et al. [1] compare AI-generated code with that produced, unaided, by students, as well as the effect of prompt variations on the code output—related topics but not our goal here. They note that "we cannot put the genie back in the bottle. The question arising for the computing educating community … is how we engage with the challenges and opportunities presented by the by the increasing effectiveness of machine learning tools ..." Finnie-Ansley et al, along with many researchers, suggest having students critique the output of AI generators. AI generation critique is somewhat incorporated in our approach in that students must inspect AI-generated material before they can add value to it.

Using chatGPT to *create* assignments has been discussed by several researchers (a recent example is Zhai [6]), but, while related, this was not our goal. An early use of AI in education has been to grade, or assist in grading, papers (e.g., Stanyon et al [3],

[3] We are indebted to Warren Mansur for this prompt example, slightly edited.

Restrepo-Calle et al [4], and Zhai [6]). Many researchers have investigated the *detection* of cheating via AI generation. So far, this has had mixed results (e.g., Khalil and Er [5]). Plagiarism detection is not our approach, however.

Finnie-Ansley et al suggest placing "more emphasis on code review, or evaluating code." This is consistent with our view and approach. They ask the question "How can an educator differentiate between a student who tried, but ultimately failed to get the correct answer and a student who simply generated code using (an AI)?" Our work deals with this question.

## 3 Approach

Students in the spring 2023 Systems Analysis and Design and Advanced Programming courses were required to leverage AI generation as much as they could—to "push" it. They could request to opt out with justification, however, which we told them was likely to be approved. Prompt engineering examples were provided as in Sect. 1.1. We emphasized specificity, focus, and hyper-prompting. *Specificity* includes being explicit about the word counts and literacy level of outputs, requesting a format such as numbering, and using template-like underscores in prompts. *Focus* concerns applying one aspect at a time, although this is not always desirable. *Hyper-prompting* includes asking chatGPT for effective prompt suggestions.

Assignments were in essay or project form (in particular, not multiple choice). The assignment grading criteria were based on students' value-added, as follows for the systems analysis class:

1. *Your contributions to clarity*
2. *Your contributions to technical soundness*
3. *Your contributions to thoroughness & coverage and*
4. *Your contributions to relevance*

For the advanced programming class, the criteria consisted of contributions to clarity, implementing requirements, and technical correctness.

Student work was evaluated via their prompt sequences, together with their edits—including additions—to chatGPT's output produced by their prompts. Because this material can be voluminous, the student's summary of their value added was required to be listed first, the remainder acting as documentation of the claims made.

Students were provided with an MS Word template for their assignments that included sections for their responses to specific topics. We tried two formats for this, mindful of the large resulting quantity of text. The first template format, shown in italics next, supplemented the template with an MS Word 'note' of variable length, where students were requested to provide their prompt sequence and the text resulting from it.

***Class Model***
*Provide a class model for the system, maintaining the system scope you determined. Your solution should have roughly 12 classes. (When complete, a real design typically may contain hundreds of classes, but your submission must focus on the scope you have chosen.) Your class model should show classes and their relationships. To add clarity to*

*your diagram, provide key attributes and methods. (You do not have to list every attribute and method, just the most important ones.) Label clearly.*

*Your class model and notes replace this.*

*Replace this with the chatbot text from which you edited your response or with "no AI used for this part."*

*MY SEQUENCE OF PROMPTS:*

*(Expand this text box downward to work on, but please restore it to just the first line when done.)*

*1.*

An improved format provided an ordinary MS Word headings for this purpose because these are collapsible, and this solves the volume problem. The usual plagiarism rules apply to prompts: copied ones had to be acknowledged.

## 4 Results

Besides describing methods used to deal with AI generation, this study includes three small surveys in the Spring 2023 Systems Analysis class. The purpose of the latter was to frame hypotheses, to suggest measurements in future studies, and to learn what questions to ask in a large-scale investigation.

Every student in the Systems Analysis class and, at the time of writing this, the Advanced Programming class, opted to use AI generation on every assignment, despite having the option to avoid it. About four of the sixteen students in the former class had not tried chatGPT when the class began. Initially, only a small minority in that class felt comfortable leveraging AI generation to the maximum extent.

Identifying survey question phraseology in the AI-assisted environment was one purpose of our study. Students were surveyed three times in the Systems Analysis class, using similar questions. The first survey was conducted two weeks after the seven-week course began; the question was "How effective is the integration of chatbots into the assignments?" The results are shown in Fig. 1. The second survey was conducted four weeks after the course began. The question was "How helpful or unhelpful for your learning is the option to use a chatbot for assignments (compared with having it prohibited)?" The results are shown in Fig. 2 The third survey was conducted six weeks after the course began, the question being "How helpful or unhelpful for your learning is the option to use a chatbot for assignments (compared with having it prohibited)?" The results are shown in Fig. 3.

The final exam in the systems analysis course was proctored, and AI generation was prohibited. There were no significant difference in grades when comparing the finals with the assignments.

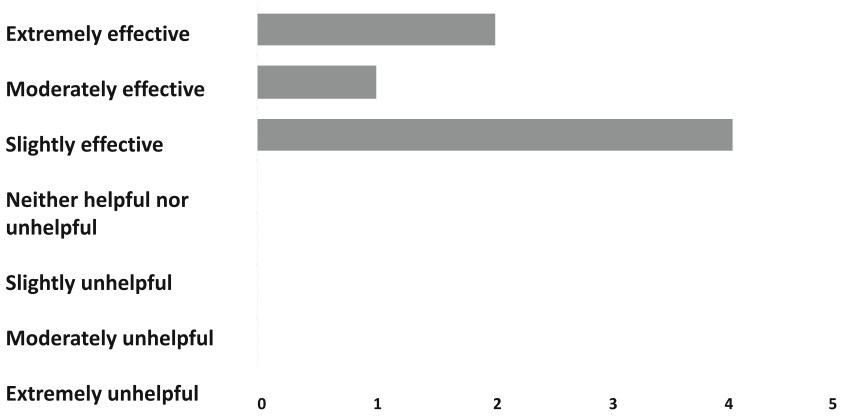


**Fig. 1.** Results for "How effective is the integration of chatbots into the assignments?" (week 2)

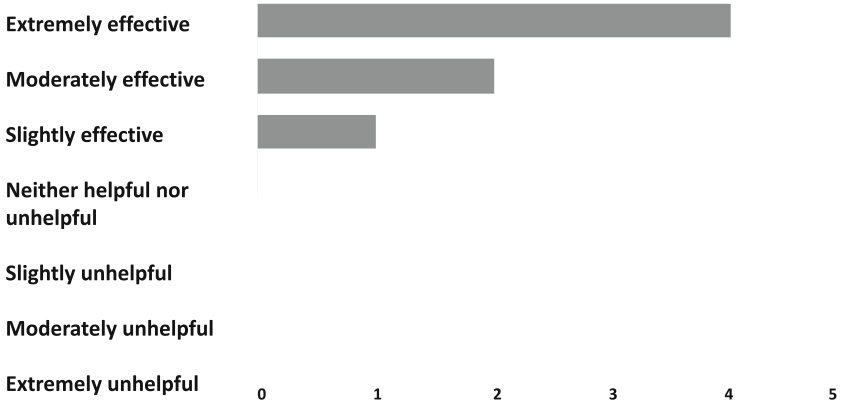


**Fig. 2.** Responses to "How helpful or unhelpful for your learning is the option to use a chatbot for assignments (compared with having it prohibited)?" (survey 2, week 4)

## 5 Analysis

This study concerns the learning that students experience in the AI-generation era and the design of assignments to promote such learning. Our approach, *value-added*, and students' reaction to it—and to using AI generation—was the specific subject.

Students showed an interest in using chatGPT throughout both courses. In the beginning, several students submitted work with only minor edits to AI-generated output, but this changed as we emphasized that credit would be given only for their value added and as they became accustomed to appropriately leveraging chatGPT. We don't want a student to edit just for the sake of editing, however: either the grader can show that added value is possible—otherwise the student's response contributes neither way except insofar as the prompt is added value.

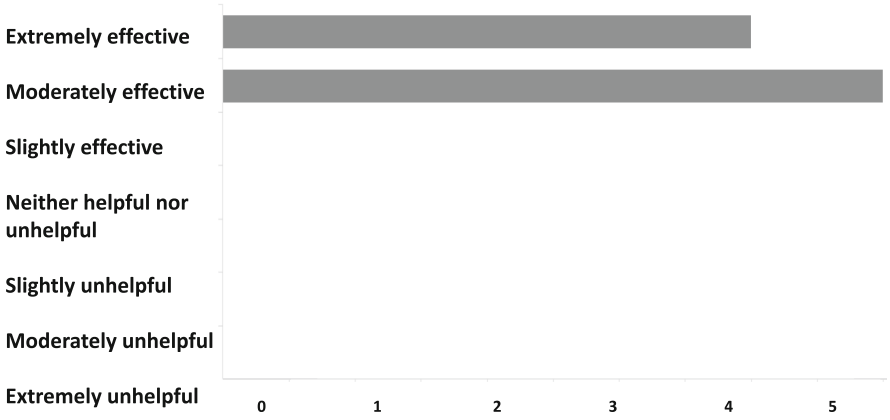


**Fig. 3.** Responses to "How helpful or unhelpful for your learning is the option to use a chatbot for assignments (compared with having it prohibited)?" (survey 3, week 7)

As Figs. 1, 2, and 3 show, there was evidence of a growing appreciation among students for the value of AI generation in the systems analysis class.

Although a formal comparison is yet to be performed, we were encouraged by the apparent consistency of grades when compared with the proctored, non-AI-assisted final exam in the systems analysis class. This suggests that over-reliance on AI generation may have been mitigated.

## 6 Conclusion and Future Work

The experiment suggests that students may welcome AI generation as a permanent part of their education and professional practice. The results encourage the hypothesis that the careful adding of value to AI-generated material promotes learning. The small sample size and experiment informality, however, do not *prove* these conclusions, and a side-by-side evaluation comparing our approach with a credible alternative and significant number of students remains to be performed.

When introducing chatGPT-assisted work in the cited courses, we avoided discussing prompting itself (prompt engineering) in a dedicated module, fearing that this would detract too much from the course subject matter. Instead, we disseminated information such as that in Sect. 1.1 throughout the course. This is not unreasonable, but a tutorial on the use of chatGPT might have saved time discussing the tool.

Compared with evaluating students' products alone, inspecting their prompts and edits increases grading time. Primarily reviewing students' own accounts of their value added mitigates this time problem. But the past may be irretrievable in any case: AI generation makes it possible to generate much greater volume, which will take more time to evaluate in any case.

As prompting becomes increasingly important, the clear use of (natural) language increases in importance for students in technical fields like systems analysis and software development.

More fundamentally, the use of AI generation means that students must learn to operate, academically and professionally, at a higher level than heretofore. Rather than think like a "classical" individual contributor, students and practitioners will function, in effect, with the help of an eager, ever-present assistant who is extremely knowledgeable in, and very good at integrating and expressing specifics that have been created by others. In other words, they will operate at a "contributing technical managerial" and "contributing editorial" level.

Critical thinking is key to dealing with chatbot output: AI generation encourages teaching and learning at the *evaluate* end of Bloom's taxonomy [7]: *remember*, *understand*, *apply*, *analyze*, *evaluate*, and *create.*

We propose that the approach to incorporating AI generation in learning described in this paper forms a reasonable basis for moving forward in various IT and computer science courses.

**Acknowledgement.** The author is indebted to the facilitators in the two classes mentioned who evaluated, or are evaluating, student assignments with the approach described here: Pamela Farr, Kuang-Jung Huang, Tony Lucarelli, and Warren Mansur.

## References

1. Finnie-Ansley, J., Denny, P., Becker, B., Luxton-Reilly, A., Prather, J.: The robots are coming: exploring the implications of OpenAI codex on introductory programming. In ACE 2022: Proceedings of the 24th Australasian Computing Education Conference, pp. 10–19 (2022). https://doi.org/10.1145/3511861.3511863
2. Floridi, L., Chiriatti, M.: GPT-3: its nature, scope, limits, and consequences. Mind. Mach. **30**(4), 681–694 (2020). https://doi.org/10.1007/s11023-020-09548-1
3. Stanyon, R., Martello, E., Kainth, M., Wilkin, N.K: Demo of Graide: AI powered assistive grading engine. In: L@S 2022: Proceedings of the Ninth ACM Conference on Learning @ Scale, pp. 466–468 (2022). https://doi.org/10.1145/3491140.3528263
4. Restrepo-Calle, F., Ramírez-Echeverry, J., Gonzalez, F.: Uncode: interactive system for learning and automatic evaluation of computer programming skills. In: EDULEARN18 Proceedings, pp. 6888-6898 (2018). https://doi.org/10.21125/edulearn.2018.1632
5. Khalil, M., Er, E.: Will ChatGPT get you caught? Rethinking of plagiarism detection. arXiv: 2302.04335 (2023)
6. Zhai, X.: ChatGPT for next generation science learning. crossroads. ACM Mag. Stud. (XRDS) **29**(3), 42–46 (2023). https://doi.org/10.1145/3589649
7. Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R.: Taxonomy of Educational Objectives: The Classification of Educational Goals. V. Handbook I: Cognitive domain. David McKay Co., New York (1956)
8. Sadasivan, V., Kumar, A., Balasubramanian, S., Wang, W., Soheil Feizi, S.: Can AI-generated text be reliably detected? arXiv:2303.11156v1 (2023) https://doi.org/10.48550/arXiv.2303.11156