POLITECNICO DI TORINO
Repository ISTITUZIONALE

Is ChatGPT Capable of Crafting Gamification Strategies for Software Engineering Tasks?

Original
Is ChatGPT Capable of Crafting Gamification Strategies for Software Engineering Tasks? / Fulcini, Tommaso; Torchiano, Marco. - ELETTRONICO. - (2023), pp. 22-28. (Intervento presentato al convegno Gamify '23: 2nd International Workshop on Gamification in Software Development, Verification, and Validation tenutosi a San Francisco (USA) nel December 4, 2023) [10.1145/3617553.3617887].

Availability:
This version is available at: 11583/2981212 since: 2023-08-23T15:15:42Z

Publisher:
ACM

Published
DOI:10.1145/3617553.3617887

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

01 August 2024

Is ChatGPT Capable of Crafting Gamification Strategies for Software Engineering Tasks?

Tommaso Fulcini
Politecnico di Torino
Turin, Italy
tommaso.fulcini@polito.it

Marco Torchiano
Politecnico di Torino
Turin, Italy
marco.torchiano@polito.it

ABSTRACT

Gamification has gained significant attention in the last decade for its potential to enhance engagement and motivation in various domains. During the last year ChatGPT, a state-of-the-art large language model has received even more attention both in the field of scientific research and in common use by individuals or companies.

In this study, we investigate the possibility of adopting ChatGPT as a tool for designing gamification platforms in the Software Engineering domain. Leveraging the capabilities of ChatGPT, we assess how good is it at generating effective suggestions and ideas for designers or developers.

To evaluate ChatGPT's potential as a gamification platform creator we narrowed the context to one particular Software Engineering activity, asking for possible aspects of the activity to be gamified. Each proposed aspect was subsequently unraveled by ChatGPT both asking in a shared and separate context, first following the conversational nature of the model, then applying a validated design framework. The study assesses ChatGPT's ability to select and integrate game elements to build a thriving gamification environment by framing the design of the platform to a state-of-the-art conceptual framework. To evaluate the goodness of the design choices made we relied both on the Octalysis framework and on personal experience.

The findings of the papers show that ChatGPT can only create simple playful experiences not very effective. Although, by instructing the model with more specific desired mechanics and dynamics, it is possible to guide it toward the application of the ideas suggested. We argue that ChatGPT is not capable of building a gamified environment on its own, but it could still be used to build the foundation of a gamification platform as long as the designers refine and rough out the advice gained from a user-centered solution.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; • **Software and its engineering** → **Software creation and management**.

KEYWORDS

Gamification, Software Engineering, Software Lifecycle, Large Language Model, Artificial Intelligence

ACM Reference Format:
Tommaso Fulcini and Marco Torchiano. 2023. Is ChatGPT Capable of Crafting Gamification Strategies for Software Engineering Tasks?. In *Proceedings of the 2nd International Workshop on Gamification in Software Development, Verification, and Validation (Gamify '23), December 4, 2023, San Francisco, CA, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3617553.3617887

This work is licensed under a Creative Commons Attribution 4.0 International License.

Gamify '23, December 4, 2023, San Francisco, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0373-7/23/12.
https://doi.org/10.1145/3617553.3617887

1 INTRODUCTION

During the last decade, a positive trend in the use of gamification techniques in several domains can be observed. Its wide adoption spans from marketing to education [19] [16], from healthcare to software testing [2] [10]. Gamification is a technique that can be defined as *"the use of game design elements in non-game contexts"* [6] and its main objective can be the improvement of users' efficiency or effectiveness at performing particular tasks, the promotion of a particular habit, or even the increase of in engagement of the gamified activity.

Although gamification can bring several positive aspects, this technique is not exempt from flaws and drawbacks due to the high dependence on the human factor [18] [10]. Developing a successful gamified environment is a complex task, which largely depends on both the type of activity to be gamified and the features of the involved users.

Since the release of ChatGPT, dated back in November 2022, the usage of this kind of Large Language Model (LLM) has spread from just IT experts to common people, reaching the amount of one million users in only 5 days [8], mostly thanks to the textual interface enabling users to interact with the Artificial Intelligence directly using the Natural Language (NL).

Given ChatGPT's ability to quickly provide implementational solutions that fit into a context that is made known to it, the question arises as to whether or not ChatGPT is capable of dealing with a task as complex as the development of a gamification platform. In particular, the design phase is the most critical, being subject not only to structured methodologies but also to a creative component that cannot be attributed to machine learning algorithms by their definition.

The massive popularity received by ChatGPT in the last months has led the research community to question the adoption of this tool in several domains of application, other examples of its usage are available in the literature [13].

In this paper, we will address this question to design a gamified platform that handles one Software Engineering activity with the aid of ChatGPT. The remainder of the paper is organized as follows: in Section II we provide some background regarding gamification principles and large language model, Section III illustrates the methodology we followed to perform our investigation, Section IV provides a discussion of our findings, and finally, Section V concludes the paper with a final resume and some remarks for all those who will approach its use in the future.

2 BACKGROUND

2.1 Gamification In Software Engineering

To approach the topic of gamification is necessary to introduce a rigorous vocabulary which, however, is not uniformly and consistently adopted in the literature. In this paper, we will stick to the definitions proposed in the MDE framework by Robson et al. [23] distinguishing between game mechanics, game dynamics, and emotions.

Delving into the definitions, *game mechanics* are the basic building blocks that form a game experience, the designer defines them and are independent from player to player, i.e. for a particular game experience they are fixed once. Examples of mechanics are points, achievements, and leaderboards. Whilst, *Game dynamics* are the desired behaviors that emerge during play: they are dependent on how players follow the mechanics set by the designer. *Emotions* are the mental states that players perceive during the gamified experience, i.e. being the product of how mechanics are followed and the dynamics that have been generated. The task of a designer is to select and integrate the game mechanics that best suit the context to be played, to stimulate the dynamics that best trigger the desired emotions.

In the context of Software Engineering, a design framework for creating gamified experiences has been proposed by Dal Sasso et al. [5]. The framework is based on *Activities* that are to be gamified. Each activity is decomposed into three aspects:

1. *Analysis*, which presents the rationale and the emotional goals to be reached.
2. *Implementation*, which includes actors involved in the activity, dynamics they are subjected to, possible hazards that may occur and game mechanics, which Dal Sasso et al. refer to as metas. In our paper we will refer to this aspect as game mechanics, adopting a stricter nomenclature.
3. *Testing*, which includes the target entities, i.e. metrics, the methodology that can be used to perform the testing, and lastly both the expected and the actual results.

Another framework that allows characterizing the produced playful environment based on the selected game mechanics, the way they can be implemented, their effects, and their relations is the *Octalysis* developed by Yu-kai Chou [4]. The framework identifies eight *Core Drives*, which are various dimensions of gamification mechanics that target different aspects of players' engagement (such as rewarding, social interaction, and ownership).

To emphasize the importance of balancing and integrating different categories, he divided them into two main groups. The first group, known as "Right Brain"/"Left Brain", distinguishes mechanics that are based on logical reasoning from those based on emotional aspects. The former is more commonly used due to its ease of implementation and quick feedback; however, overuse can lead to stagnation and drive away users. The latter is more delicate as it must be tailored to suit the involved users to bond them with the environment.

The second group, "Black Hat"/"White Hat", focuses on the emotions the mechanics evoke in users. White Hat emotions are the positive feelings that the user wants to achieve while playing. Instead, the negative outcomes that users should avoid in the environment are referred to as Black Hat. Some mechanics may yield quick results but have negative long-term effects, while others provide more balanced and positive sensations in the long run.

Gamification techniques have been applied in several activities of Software Engineering, spanning from Requirements Engineering to Software Testing [24] [11], both for conducting and for teaching the target activity [9] [3]. The degree of maturity in this research area has grown to the point where a tertiary study was published in 2020 [12], with other newer secondary studies released in later years [20] [10] [17].

2.2 Large Language Models

LLMs are a family of Artificial intelligence (AI) techniques that are trained with a large amount of textual data with a huge amount of parameters. LLMs are particularly suitable to decode structures and patterns of NL, some state-of-the-art examples of pre-trained language are GPT (Generative Pre-trained Transformer) [21], T5 (Text-To-Text Transfer Transformer) [22], and BERT (Bidirectional Encoder Representations from Transformers) [7].

GPT 3.5 (the model which ChatGPT relies on) has shown a high capability of In-Context Learning (ICL), i.e. the model's feature to understand and provide customized answers based on an input context, instead of purely relying on internal knowledge of the model obtained during the pre-training phase [14]. Another technique used in LLMs is Chain of Thought (CoT) [26] that provides multiple demonstrations to describe the CoT as examples within the prompt, guiding the model's reasoning process. The concept of self-consistency extends the technique of CoT [1] [25], where a majority voting mechanism on the answers allows to reach higher reliability, keeping consistency throughout the conversation between the AI and the user.

3 RESEARCH METHODOLOGY

Our research objective is to verify whether ChatGPT can be used as a means to suggest effective gamification strategies in the context of Software Engineering. To do so, we queried the model with the conversational interface using the following methodology.

To address the viability of using ChatGPT as a design assistant, we started by randomly extracting one target Software Engineering activity: to this extent, we asked ChatGPT itself to list all the known activities and to randomly select one of them. The choice fell on Debugging and troubleshooting activity.

Once we extracted the target process for which to design a gamified environment, we followed two distinct paths: first, we tried a boundless approach, secondly we decided to rely on a more strict methodological design framework. We define the **boundless** approach as *a free question-and-answer approach in which the user deepens the design aspects of the platform to build, making the ideas and their implementation emerge throughout the conversation.* The adoption of this methodology was mainly driven by the conversational nature of ChatGPT, which allows AI to bring out the details of the proposed solution when requested. We define the **structured** approach *as an iterative question-and-answer approach in which the user requests aspects that stick to particular frameworks with ad-hoc definitions*. To this extent, we selected the design framework provided by Dal Sasso et al. as a reference methodology to formalize the design choices [5]. To follow this latter approach it was necessary to instruct the AI about the design framework, as it was not able to produce answers compliant with it by default.

For both approaches, all the questions asking for design solutions were asked twice: once using *shared context* queries, and once using *separate context* queries. We define a **shared context** as *a query scenario, where all sub-questions are asked in one ChatGPT session in one single conversation, relying upon the ICL and CoT features of ChatGPT*. Whilst, we define **separate context** as *a query scenario in which each subject to be gamified is asked within a separate and independent conversation*. The above definitions are adapted from Jalil et al. in [15] to fit our application context.

The *shared context* allows having short prompts which are refined progressively by adding more details in subsequent inputs, constructing the information step by step. Conversely, the *separate context* requires the input to be more precise and verbose, enveloping the context, the methodology, and the request in one single input. Starting from several inputs used in a *shared context* it is possible to construct the corresponding input to be used in the *separate* one, although the outputs are far from equivalent, given the stochastic nature of the model.

We finally evaluated the outcomes in two ways: first with the Octalysis framework, to characterize the proposed solution from a game design point of view, relying also on the score (that spans from 0 to 800) of evaluation and comments that arise when the corresponding octagon is shaped in the official website[1]. These results are plotted in the Octalysis in Figure 1. Secondly, assessing the adherence to Dal Sasso and MDE frameworks, mainly relying on our experience in the field. A summary of this evaluation can be found in Table 1.

The original complete sequences of questions and answers are publicly available [2].

4 RESULTS

In this section, we will describe separately what we obtained from the queries made, and then compare and analyze the results obtained.

4.1 Boundless Approach

4.1.1 Shared Context. The querying process started by asking the model how to create a gamified environment in the context of debugging and troubleshooting. The first responses suggested features mainly related to the training aspect of the context with playful tools to learn how to debug and maintain the code. We then instructed the model to consider a real scenario in which the people involved use the gamified platform to perform their daily job, e.g. a consulting company. From this input, ChatGPT extracted a list of possible features to implement, which will be also used to ask the questions to the model in the *separate contexts*.

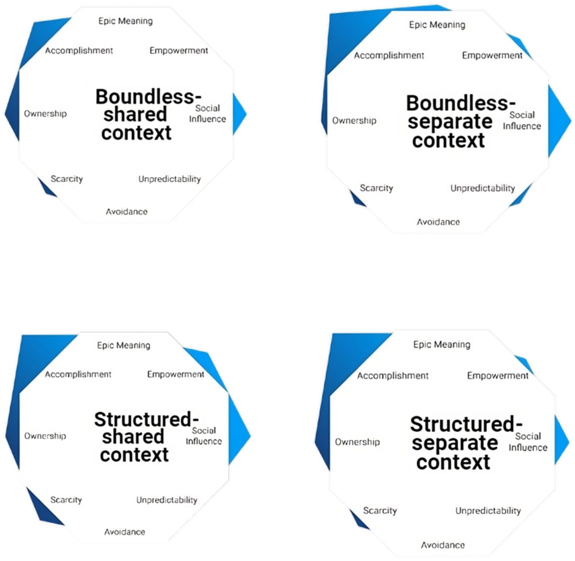

Figure 1: Comparison of Octalysis evaluation of the gamified environment proposed by each approach

We asked the model to elaborate more on each item of the list, identifying which were the basic game mechanics to implement, the rules that govern them and the feature of the activity that had to be involved. At this point, the model started providing contradictory responses, suggesting both competitive and collaborative-based mechanics, confusing features with game mechanics and vice versa. By clarifying the concept of rules, game mechanics, dynamics and technical features the model improved a bit the accuracy, but partially lost the bigger picture given by the context described in the first questions, returning to suggest proposals devoted to learning instead of conducting the activities.

Upon closer examination of the game mechanics, we noticed that the suggestions consistently pertain to scores, rankings, and achievements. Additionally, the game incorporates time-based competitions, rewards, and generic challenges. It is important to ensure that these elements are properly assigned and not confused with other artifacts. It is worth noting that a focus on collaborative activities was registered in the answer, even if no idea emerged about how to make such activities cooperative.

4.1.2 Separate Context. Querying ChatGPT using *separate contexts* requires more thorough questions than in the *shared context*, for this reason, we condensed in only one question the description of the application context and the subject to be gamified taking each item from the list.

Although in most cases the model misclassified game mechanics, rules, and features, it generated some fascinating outputs. The rules it suggested often included cooperative elements that aimed to

[1] https://yukaichou.com/octalysis-tool/
[2] https://www.doi.org/10.6084/m9.figshare.23709747

produce high-quality work, promote player progress, manage time effectively, and offer rewards for achievements.

After providing an unambiguous description of what game mechanics, dynamics, rules and technical features are, the model reacted by refining its classification of the proposed elements. However, the output started becoming a bit contradictory as what happened in the *shared context*, such as an unjustified cooperative/competitive dichotomy.

We finally tried pointing out that the proposed elements for a gamification platform were just the basics and that more complex dynamics were needed. We asked for suggestions to set up more complex dynamics, the model generated a list of items related to debugging and troubleshooting. Unfortunately, most of the items were either approaches for learning about the topic or elements that did not fall into the definition of game dynamics. Repeating the same question but making some examples of what more complex dynamics are (e.g. push your luck, betting, auctions), the model made suggestions that fit the provided examples, applying the correct definition of the specified dynamic but only with generic advice.

4.2 Structured Approach

4.2.1 Shared Context. Following this approach we started from scratch selecting the target sub-activities and features to gamify. Once obtained the list, we needed to instruct the model with the context of the application and the methodological framework to follow. A first attempt was made by presenting both in a single input, but the resulting output was lacking in adherence to both the context and the methodological framework. We then proceeded to ask the model to perform the design activity for each activity, proceeding "horizontally" considering one layer of the framework for each activity at a time.

Each time we queried the model on an activity in the list, we reported a summary of the output produced previously, to avoid responses with low internal consistency due to the possible information overload of considering several activities at the same time.

Querying ChatGPT in this way produced appropriate output for the Analysis layer, listing good rationales and emotional goals for most of the activities. The implementation layer is the one lacking the most, as the precision of the model at correctly classifying game dynamics is still pretty low, and the variety of game mechanics suggestion is low as well. Nonetheless, both the hazards found were actual risks to be raised and the Testing layer proposed good metrics and methodology for verifying the implemented solutions. We consider the generated expected results as too optimistic and peremptory, as ChatGPT only highlights positive results.

4.2.2 Separate Context. Using the *separate context* querying approach we needed to summarize several questions in only one, as we could not rely on ICL. We started each conversation with one question aimed at instructing the model with the general application context, the sub-activity of Debugging to consider, and the general purpose of creating a gamification platform.

One first difference with the *shared context* is that the model always listed more stakeholders than were actually needed. This overestimation is actually repeated for many of the questions asked. Additionally, in some cases, ChatGPT assigned the wrong roles to some stakeholders, failing at identifying the actual target of the activity.

Considering the analysis layer we note that the output obtained was rather generic. In fact, the rationale was almost repeated each time, being mostly referred to the general platform instead of focusing on the specific task at hand. This append to the emotional goal, which was repeated as well, involving aspects of cooperation, excitement, and satisfaction.

The implementation level, instead was more precise at correctly identifying game dynamics and mechanics even though little or no clue on how to integrate all the proposed elements was actually given. On the other hand, the hazards were successfully identified by the model, proposing plausible threats that may occur in the gamified environment.

Finally, considering the testing layer ChatGPT overestimated the number of metrics to measure, assessing dimensions related to other debugging sub-activities, ending up out of context in some cases. The methodology to be employed sometimes lacks feasibility, e.g. continuous monitoring, and adherence with the case of study e.g. surveys to collect the feedback which is difficult to operate assuming a continuative use of the platform. As in the shared context, we felt the expected results were too optimistic.

A summary of the obtained results can be seen in Table Table 1.

5 DISCUSSION

Upon examining the obtained output, one might at first glance think that ChatGPT could be able to recognize the application context of gamification and apply playful elements to it. However, while the solutions proposed may describe ways to implement gamification platforms, their effectiveness has yet to be proven.

Once created the Octalysis of the gameful environment proposed by each approach, we gathered the scores and the comments from the website. In all four cases, the evaluation was negative, highlighting the lack of identity in the design and classifying the combination as a weak experience that is unable to produce desired outcomes.

As it can be seen, Figure 1 shows that the proposed game elements are mainly polarized in the *accomplishment* core drive. This trend in the model builds an unwanted situation where the game elements create an unbalanced playful experience. The result of this is the rapid abandonment of the platform by users. Additionally, this polarization is neither strong enough to avoid a shallow experience: this situation where there is neither balance nor a strong identity in any of the four dimensions gets the designer nowhere.

Considering the adherence to MDE and Dal Sasso Framework, we verified that the model by default has very low accuracy in classifying the output. Even when the concepts are made explicit throughout the conversation the model often makes classification mistakes. None of the applied paths can be unanimously considered better. We would nevertheless advise anyone approaching the use of ChatGPT as a design process assistant to use the structured approach for more consistent results. We summarized in Table 1 the obtained output, mapping the results to Dal Sasso's Framework.

We contend that the proposed design solutions are way too naive to rely upon, as none of the querying paths was able to produce an environment being both satisfactory and comprehensive of the starting problem. We emphasize that creating a successful

Table 1: Summary of the findings of each applied methods mapped to the used design framework

Dal Sasso's Design Framework Layer	Boundless-Shared Context	Boundless-Separate Context	Structured-Shared Context	Structured-Separate Context
Analysis	The model suggested mainly educational aspects, providing no clues about the rationale.	The model identified the simple but sound logic, aiming at enhancing enjoyment and fostering collaboration.	The model set the emotional goals of empowerment and pride for the work done and collaboration between people with the rationale of enhancing user's ability and satisfaction.	The model recognized as rationale the performance improvement with the emotional goal of promoting collaboration between stakeholders with a sense of satisfaction.
Implementation	Only the main actor was identified, the model had very low accuracy in proposing dynamics and mechanics providing only the basic ones. Hazards were never mentioned.	Only the main actor was identified, a broader set of mechanics was suggested, mainly based on accomplishment, with an accurate description of how to implement them. Only simple and conflictual dynamics emerged, without any mention of potential risks.	The model correctly identified the actors and suggested a wide set of mechanics, mainly related to left-brain gamification elements, a special focus was put on collaboration. The model warned about potential common risks that may arise.	The model overestimated the actors, involving too many of them. A wide set of game mechanics was proposed, not deviating from the base set of left-brain gamification elements. The main dynamics proposed were collaboration, competition, and progression, potential drawbacks were correctly listed.
Testing	The model did not generate any information related to the way to be used to evaluate the gamified environment produced. Only some generic potential benefits were highlighted as potential outcomes.	The model did not propose any method to evaluate the environment. Some potential benefits have been generated as hypothetical outcomes from the use of the platform.	The model suggested proper metrics and methodology for evaluating the outcome of applying the playful environment, except for a few sporadic cases it suggested evaluation methods that are unfeasible in the way the platform is implemented. However, the expected results produced are too optimistic.	The model generated an overly broad set of metrics, including measurements out-of-scope. The methodology to employ was realistic, while the expected outcomes were overly estimating benefits, assuming overly optimistic results.

gamification environment is a complex process that involves not just the activity being targeted but also the people who will be using it.

It's expected that ChatGPT can provide suggestions based on its training. Therefore, it is not surprising that the gamification elements recommended by ChatGPT match the most widely spread in the literature. However, we know that the success of an implemented solution does not depend solely on individual elements but on how well they're integrated and adapted to the specific context.

On that basis, it is crucial to adopt a human-centered design approach based on the study of the personas, tailoring the platform to their needs in lieu of following a predefined structure, which is done by ChatGPT.

In conclusion, we believe that ChatGPT, as well as other AI tools, are currently unable to generate valid solutions all on their own. However, they can serve as a fascinating and efficient way to provide feedback on assessing a proposed solution or identifying potential risks in utilizing a platform. They can also be used to improve designers' concepts, as long as the user has a clear understanding of their requirements and explicitly specifies them.

6 THREATS TO VALIDITY

In the adopted methodology, we followed what we called *boundless* and *structured* approaches which are ad-hoc procedures that we developed for the occasion. Given the novelty of ChatGPT, at present no comprehensive and clearly defined methodology is widely recognized in the research literature for researchers to adhere to.

This methodology has not yet been validated, nor are the results themselves generalizable as the output of ChatGPT is based on stochastic processes. Regarding the *separate* and *shared* context, we relied on a procedure previously used by Jalil et al. [15].

Although none of the four methodologies has clear benefits and drawbacks demonstrated, we argue that this validity threat was partially mitigated by performing the design task four times in semi-independent ways (a slight dependence was present due to the shared list of features to be developed that was necessary to compare the obtained results).

Additionally, we used two different sets of tasks to be designed, one in the *boundless*, and one in the *structured* approach: this could constitute a validity threat as, having different inputs the reasoning of the model might have been affected. However, our aim was not to compare the results themselves but to evaluate the soundness, a viable way to mitigate this threat could be blending the outputs and sieving only what the designer considers useful.

Other threats include construct validity as we admit that the whole experiment was carried out using the model GPT 3.5, which at the time of writing is free for everyone. The model GPT 4, which is recognized by OpenAI itself as their most advanced model, may have produced different yet better results compared to GPT 3.5. Since GPT 4 comes with a fee for the users, we opted for the free version to provide readers with an overview of the result that anyone can achieve simply by using Chat GPT.

Finally, we highlight an external validity threat regarding the generalizability of the results, that may not be applicable in situations where gamification is used for activities outside of Software Engineering. While we discovered that the shortcomings of ChatGPT stem from issues with game design application rather than a lack of knowledge about the target domain, it's possible that the model could be more effective in other domains with appropriate training and development. However, this does not detract from the fact that the use of ChatGPT remains more akin to imitating pre-existing solutions rather than developing new customized ones.

7 CONCLUSION

This paper explored the application of gamification in the context of software engineering, focusing on the design of gamified experiences using ChatGPT as a tool for generating solutions. For this study we developed two different approaches were used: a boundless approach and a structured approach based on a design framework by Dal Sasso et al. The results and discussion shed light on the capabilities and limitations of ChatGPT in providing gamification solutions for the specific activity of debugging and troubleshooting.

The findings indicate that ChatGPT can suggest gamification elements, such as scores, rankings, and achievements, but often lacks coherence and understanding of the larger context in which these elements should be integrated. The suggested solutions are generally naive and not comprehensive enough to create a successful gamification environment, emphasizing the need for a human-centered design approach that takes into account the specific needs and goals of the target users.

We found that ChatGPT can be valuable for providing feedback on a completed solution and identifying potential risks, but it is not yet capable of generating fully valid solutions all by itself without human supervision.

The research highlights the importance of human expertise and understanding in the design of gamification platforms. As AI continues to advance, it is essential to leverage its capabilities while also recognizing its limitations and integrating human judgment to create effective and meaningful gamified experiences in software engineering and beyond.

REFERENCES

[1] Toufique Ahmed and Prem Devanbu. 2023. Majority Rule: better patching via Self-Consistency. *ArXiv* abs/2306.00108 (2023). https://doi.org/10.48550/arXiv.2306.00108

[2] Saja Al-Rayes, Fatemah Ali Al Yaqoub, Asma Alfayez, Demah Alsalman, Fahad Alanezi, Seham Alyousef, Heba AlNujaidi, Atheer K. Al-Saif, Razaz Attar, Duaa Aljabri, Sama'a Al-Mubarak, Mona M. Al-Juwair, Sumaiah Alrawiai, Linah Saraireh, Amjad Saadah, Arwa Al-umran, and Turki M. Alanzi. 2022. Gaming elements, applications, and challenges of gamification in healthcare. *Informatics in Medicine Unlocked* 31 (2022), 100974. https://doi.org/10.1016/j.imu.2022.100974

[3] Kylie Bedwell, Giacomo Garaccione, Riccardo Coppola, Luca Ardito, and Maurizio Morisio. 2023. BIPMIN: A Gamified Framework for Process Modeling Education. *Information* 14, 1 (2023). https://doi.org/10.3390/info14010003

[4] Y.K. Chou. 2015. *Actionable Gamification: Beyond Points, Badges, and Leaderboards.* Createspace Independent Publishing Platform. https://books.google.it/books?id=jFWQrgEACAAJ

[5] Tommaso Dal Sasso, Andrea Mocci, Michele Lanza, and Ebrisa Mastrodicasa. 2017. How to gamify software engineering. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 261–271. https://doi.org/10.1109/SANER.2017.7884627

[6] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. c. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (Tampere, Finland) *(MindTrek '11)*. Association for Computing Machinery, New York, NY, USA, 9–15. https://doi.org/10.1145/2181037.2181040

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). https://doi.org/10.48550/arXiv.1810.04805 arXiv:1810.04805

[8] Fabio Duarte. 2023. Number of ChatGPT Users (2023). https://explodingtopics.com/blog/chatgpt-users Accessed: 2023-07-05.

[9] Tommaso Fulcini and Luca Ardito. 2022. Gamified Exploratory GUI Testing of Web Applications: a Preliminary Evaluation. In *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 215–222. https://doi.org/10.1109/ICSTW55395.2022.00045

[10] Tommaso Fulcini, Riccardo Coppola, Luca Ardito, and Marco Torchiano. 2023. A Review on Tools, Mechanics, Benefits, and Challenges of Gamified Software Testing. *ACM Comput. Surv.* (feb 2023). https://doi.org/10.1145/3582273

[11] Giacomo Garaccione, Tommaso Fulcini, and Marco Torchiano. 2022. GERRY: A Gamified Browser Tool for GUI Testing. In *Proceedings of the 1st International Workshop on Gamification of Software Development, Verification, and Validation* (Singapore, Singapore) *(Gamify 2022)*. Association for Computing Machinery, New York, NY, USA, 2–9. https://doi.org/10.1145/3548771.3561408

[12] Gabriel Alberto García-Mireles and Miguel Ehécatl Morales-Trujillo. 2020. Gamification in Software Engineering: A Tertiary Study. In *Trends and Applications in Software Engineering*, Jezreel Mejia, Mirna Muñoz, Álvaro Rocha, and Jose A. Calvo-Manzano (Eds.). Springer International Publishing, Cham, 116–128. https://doi.org/10.1007/978-3-030-33547-2_10

[13] Puneet Gupta, Swati Raturi, and P Venkateswarlu. 2023. Chatgpt for designing course outlines: A boon or bane to modern technology. *Available at SSRN 4386113* (2023). https://doi.org/10.2139/ssrn.4386113

[14] Jiabang He, Lei Wang, Yingpeng Hu, Ning Liu, Hui juan Liu, Xingdong Xu, and Hengtao Shen. 2023. ICL-D3IE: In-Context Learning with Diverse Demonstrations Updating for Document Information Extraction. *ArXiv* abs/2303.05063 (2023). https://doi.org/10.48550/arXiv.2303.05063

[15] Sajed Jalil, Suzzana Rafi, Thomas D. LaToza, Kevin Moran, and Wing Lam. 2023. ChatGPT and Software Testing Education: Promises & Perils. In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. 4130–4137. https://doi.org/10.1109/ICSTW58534.2023.00078

[16] Amina Khaldi, Rokia Bouzidi, and Fahima Nader. 2023. Gamification of e-learning in higher education: a systematic literature review. *Smart Learning Environments* 10, 1 (31 Jan 2023), 10. https://doi.org/10.1186/s40561-023-00227-z

[17] Rita Marques, Miguel Mira da Silva, and Daniel Gonçalves. 2023. Gamification for agile: a systematic literature review. *International Journal of Agile Systems and Management* 16, 2 (2023), 226–261. https://doi.org/10.1504/IJASM.2023.130838

[18] Bita Mirzaie Feiz Abadi, Neda Khalili Samani, Alireza Akhlaghi, Sara Najibi, and Molood Bolourian. 2022. Pros and Cons of Tomorrow's Learning: A Review of Literature of Gamification in Education Context. *Medical Education Bulletin* 3, 4 (2022), 543–554. https://doi.org/10.22034/meb.2022.350941.1063

[19] Fakhroddin Noorbehbahani, Fereshteh Salehi, and Reza Jafar Zadeh. 2019. A systematic mapping study on gamification applied to e-marketing. *Journal of Research in Interactive Marketing* 13, 3 (01 Jan 2019), 392–410. https://doi.org/10.1108/JRIM-08-2018-0103

[20] Oki Priyadi, Insan Ramadhan, Dana Indra Sensuse, Ryan Randy Suryono, and Kautsarina. 2023. Gamification in Software Development: Systematic Literature Review. In *Emerging Trends in Intelligent Systems & Network Security*, Mohamed Ben Ahmed, Boudhir Anouar Abdelhakim, Bernadetta Kwintiana Ane, and Didi Rosiyadi (Eds.). Springer International Publishing, Cham, 386–398. https://doi.org/10.1007/978-3-031-15191-0_37

[21] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018). https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[22] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551. https://doi.org/10.48550/arXiv.1910.10683

[23] Karen Robson, Kirk Plangger, Jan H. Kietzmann, Ian McCarthy, and Leyland Pitt. 2015. Is it all a game? Understanding the principles of gamification. *Business Horizons* 58, 4 (2015), 411–420. https://doi.org/10.1016/j.bushor.2015.03.006

[24] Remco Snijders, Fabiano Dalpiaz, Sjaak Brinkkemper, Mahmood Hosseini, Raian Ali, and Atilla Ozum. 2015. REfine: A gamified platform for participatory requirements engineering. In *2015 IEEE 1st International Workshop on Crowd-Based Requirements Engineering (CrowdRE)*. 1–6. https://doi.org/10.1109/CrowdRE.

2015.7367581

[25] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *ArXiv* abs/2203.11171 (2022). https://doi.org/10.48550/arXiv.2203.11171

[26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv* abs/2201.11903 (2022). https://doi.org/10.48550/arXiv.2201.11903

Received 2023-08-03; accepted 2023-08-17