RESEARCH-ARTICLE

A Comparative Analysis of ChatGPT-Generated and Human-Written Use Case Descriptions

EVIN ASLAN OĞUZ, University of Applied Sciences Bielefeld, Bielefeld, Nordrhein-Westfalen, Germany

JOCHEN MALTE KUESTER, University of Applied Sciences Bielefeld, Bielefeld, Nordrhein-Westfalen, Germany

Open Access Support provided by:

University of Applied Sciences Bielefeld

Published: 31 October 2024

Citation in BibTeX format

MODELS Companion '24: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems
September 22 - 27, 2024
Linz, Austria

Conference Sponsors:
SIGSOFT

https://doi.org/10.1145/3652620.3687800
ISBN: 9798400706226

A Comparative Analysis of ChatGPT-Generated and Human-Written Use Case Descriptions

Evin Aslan Oğuz
Bielefeld University of Applied Sciences
Bielefeld, Germany
evin.aslan_oguz@hsbi.de

Jochen M. Küster
Bielefeld University of Applied Sciences
Bielefeld, Germany
jochen.kuester@hsbi.de

Abstract

The development of comprehensive use case descriptions is a critical task in software engineering, providing essential insights for requirement analysis and system design. The advent of advanced natural language processing models, such as ChatGPT, has sparked interest in their potential to automate tasks traditionally performed by humans, including the generation of use case descriptions in software engineering. Understanding the capabilities and limitations of ChatGPT in generating use case descriptions is crucial for software engineers. Without a clear understanding of its performance, practitioners may either overestimate its utility, leading to reliance on suboptimal drafts, or underestimate its capabilities, missing opportunities to streamline the drafting process. This paper addresses how well ChatGPT performs in generating use case descriptions, evaluating their quality compared to human-written descriptions. To do so, we employ a structured approach using established quality guidelines and the concept of "bad smells" for use case descriptions. Our study presents the first attempt to bridge the knowledge gap by offering a comparative analysis of ChatGPT-generated and human-written use case descriptions. By providing an approach to objectively assess ChatGPT's performance, we highlight its potential and limitations, offering software engineers insights to effectively integrate AI tools into their workflows.

CCS Concepts

• **Software and its engineering** → **Object oriented development**; **Requirements analysis**; • **Computing methodologies** → **Natural language processing**.

Keywords

use case description, ChatGPT, requirements engineering, quality

ACM Reference Format:
Evin Aslan Oğuz and Jochen M. Küster. 2024. A Comparative Analysis of ChatGPT-Generated and Human-Written Use Case Descriptions. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3652620.3687800

MODELS Companion '24, September 22–27, 2024, Linz, Austria

ACM ISBN 979-8-4007-0622-6/24/09
https://doi.org/10.1145/3652620.3687800

1 Introduction

In the realm of software engineering, the creation of use case descriptions is a fundamental activity that drives the requirement analysis and system design processes. These descriptions offer detailed scenarios of how users interact with a system, forming the backbone for understanding functional requirements and guiding the development process. Hence, their quality directly impacts the success of software projects. However, the manual creation of use case descriptions is often time-consuming and requires a high level of expertise to ensure accuracy and completeness.

Recent advancements in artificial intelligence, particularly in natural language processing (NLP), have opened new avenues for automating various aspects of software engineering [8]. ChatGPT, a state-of-the-art language model developed by OpenAI, has demonstrated significant capabilities in generating human-like text based on given prompts. This raises the intriguing possibility of employing ChatGPT to automate the generation of use case descriptions, potentially reducing the manual effort involved and accelerating the documentation process.

Understanding the capabilities and limitations of ChatGPT in this context is crucial. Without a clear understanding of its performance, software engineers may either overestimate its utility, potentially leading to reliance on suboptimal drafts, or underestimate its capabilities, resulting in missed opportunities to streamline the drafting process. If we do not ascertain of which quality the use case descriptions generated by ChatGPT are, we risk foregoing the potential efficiency gains that AI can offer. Consequently, this lack of knowledge might force practitioners to start from scratch, thereby missing out on the benefits of AI-assisted drafting.

For evaluating how well ChatGPT performs in generating use case descriptions, ChatGPT has to be applied on different problems and the generated use case descriptions have to be evaluated with regards to consistency, completeness and quality. Another question in this context is how generated use case descriptions compare to use case descriptions written by humans.

This paper presents the first attempt to bridge this knowledge gap by offering an approach for evaluating the quality of ChatGPT-generated use case descriptions and comparing them with the human-written use case descriptions. Our approach allows to objectively assess the performance of ChatGPT, highlighting its potential and limitations. By doing so, we aim to provide software engineers first insights to effectively integrate AI tools into their workflows. These insights might guide them while drafting of use case descriptions to make the process both efficient and generate high quality use case descriptions.

In this study, the scope of creating use case descriptions is narrowed down to a single case study to make the problem feasible and

propose solutions. The quality of use case descriptions is evaluated based on established quality guidelines [1] and the concept of "bad smells" in use case descriptions in the literature [15]. These guidelines include clarity, completeness, accuracy while bad smells refer to common issues such as ambiguity, inconsistency, and missing important details, poor granularity and redundancy. By applying these metrics, we offer an objective comparison of AI-generated content to human-written descriptions.

The primary contributions of this study are twofold: First, we provide a first attempt of ChatGPT's ability to understand and generate use case descriptions from predefined requirements. Second, we present an approach to compare for evaluating the quality of ChatGPT-generated use case descriptions and compare them with human written descriptions.

The paper is structured as follows: First we will explain how AI can be used in software engineering, then introduce a case study (called stock market). After that we will introduce foundations of use cases and use case descriptions, along with quality criteria for use case descriptions in Section 2. We will then report on how ChatGPT can be used for generating use case descriptions in Section 3. In Section 4 we explain our approach for systematically evaluating ChatGPT's capabilities. Subsequently we will present the results and the conclusion and finally give insights for future work.

2 Background

2.1 AI in software engineering

With the advance of artificial intelligence technology and large language models, there have been various approaches to use this technology to improve or speed up software engineering [8]. Most prominent examples include Codex [6], a large language model that Github Copilot is based on.

Recent years have given rise to several works of applying large language models to code generation and software testing. Within code generation, code completion is used for offering the software engineers an alternative to complete the code, based on incomplete code fragments. In this context, Github Copilot has received a lot of attention and its benefits in practice are under investigation [17]. With regards to code generation, various systems have been developed (including AlphaCode [10]) that takes a problem statement as input and produce a program that solves the problem as output. In software testing, large language models are used for generating new tests for programs [16].

So far, the discipline of requirements engineering has received less attention [8]. Existing work includes search-based requirements optimisation [18]. Marques et al. [12] recently provided a comprehensive review of using ChatGPT in requirements engineering. They distinguished different aspects of using ChatGPT, including effectiveness of generating requirements, the role of human input, exploration of prompt engineering and AI challenges and limitations. Recent work by Bencheickh et al. has applied ChatGPT for generating requirements [3]. Here, based on a given problem description, functional and non-functional requirements were generated and these requirements were compared with the human written requirements. The study showed that ChatGPT is able to generate proper software requirements but still falls behind humans with regards to certain quality attributes. ChatGPT has also been applied to generate user stories as inspirational triggers [11] or for evaluating user stories [14].

Since the focus of this paper is on use case descriptions, the case study that will be used to create use case descriptions will be presented in the next section. After that, use cases and use case descriptions will be presented leveraging the case study together with quality criteria.

2.2 Stock Market Case Study

Stock Market case study is one of the case studies that is being used in the System Development - Software Design course at Bielefeld University of Applied Sciences as a part of teaching material. To the best of our knowledge, the case study is not included in any published textbook and is not publicly available. It will be used throughout this study and for evaluation and is related to a system for capturing and evaluating stock investments. The description is as follows:

> A system for capturing and evaluating stock investments is required. For the use of the system, it is mandatory for users to log in. Users can register in the system and must provide a username and password, as well as a first and last name, email address, address, postal code, and city.
> A user can own any number of depots in the system, each depot is assigned to exactly one user. A depot is identified by a unique depot number, has an arbitrary name, and a total value calculated and stored by the system. Depots can contain any number of stocks and transactions.
> For stocks, information about their quantity, current price, designation, dividend yield, ISIN identification number, annual dividend, purchase price, WKN code, interest income, and currency are stored. For currencies, possible abbreviations (such as "EUR" for Euro) are stored. In addition, the associated depot for the stock is stored in the system. Different types of stocks should be supported, e.g. stocks, funds, ETFs. Stocks are divided into different categories and must be assigned to exactly one. A category always consists of a unique category ID and a designation.
> Transactions contain a unique transaction number, the time, the type of transaction (e.g. purchase or sale), the transaction value with currency (positive/negative depending on purchase or sale), the associated depot, and the bought/sold stock and its quantity.

Given such a textual description of the case study, typical first steps in system analysis and design include the derivation of a use case diagram together with use case descriptions which we will explain in the following.

2.3 Use Cases and Use Case Descriptions

Use cases are used as one of the prominent approaches for capturing requirements by software engineers. A use case describes a "...behaviorally related sequence of transactions in a dialogue between a user and the system" [9]. The Unified Modeling Language supports use case modeling by a Use Case Model which shows an overview

of use cases of a system, together with actors. An example use case diagram can be seen in Figure 1 displaying the use cases of our stock market case study, containing use cases such as *Register User*, *Create Portfolio* or *Perform Transaction*.

Details of a use case are commonly described by using a use case template. Such a template can be used to capture important aspects of a use case, such as a use case name, the goal of the use case, actors involved in the use case and the sequence of actions between an actor and the system. There is currently no standard for the use case template and there exist different forms. In this study, the one developed by Cockburn [7] will be used. An example use case description named *Perform Transaction* written following this template from our stock market case study can be found in Table 1.

Describing use cases with use case descriptions has been a manual task performed by software engineers but can be possibly sped up using AI technology which we will explore in this paper.

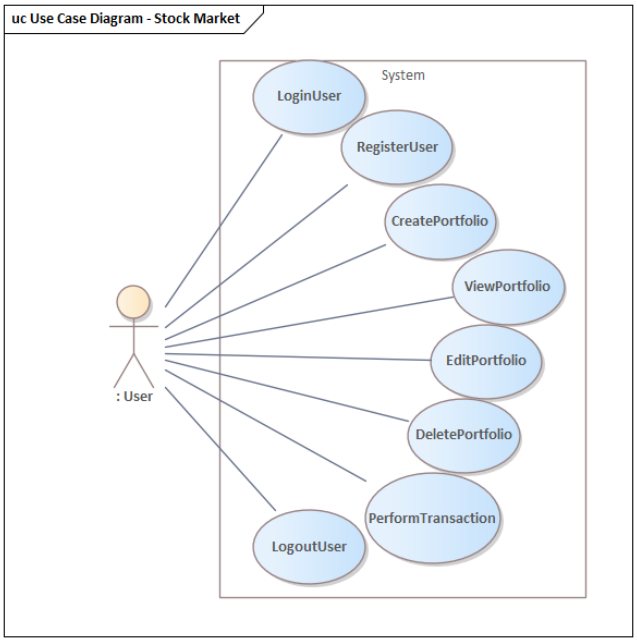

Figure 1: An example use case diagram from stock market case study

2.4 Quality criteria for use case descriptions

Use case descriptions, either created manually or by AI technology, must have a certain quality. Since use case descriptions are written using natural language, this can lead to ambiguous, inconsistent and incomplete descriptions. There have been various attempts to define guidelines on writing high quality use case descriptions. For instance, the CREWS approach [1] comprises style and content guidelines for use case authoring. An example for a style guideline is to "use present tense and active voice when describing actions". Further approaches elaborate on guidelines and provide patterns [2].

Another attempt to improve the quality of use case descriptions is to use a catalogue of bad smells [15]. The idea is to find such bad smells in use case descriptions and then remove them. Another usage of bad smells is to avoid them when writing use case descriptions. Bad smells are categorized into characteristics such as ambiguity, incorrectness, granularity, redundancy, missing information and misplacement. An example for a bad smell in the category of granularity is "Long Sentence". It describes the usage of a very long sentence in the use case description.

In the following, we will first explain the procedure that we utilized ChatGPT for generating use case descriptions. Then, we will describe our approach for evaluating the quality of human-written use case descriptions with the ChatGPT generated use case descriptions.

Table 1: A sample use case description: Perform Transaction from stock market case study.

Name:	PerformTransaction
Primary User:	User
Goal:	Perform a new transaction in the system
Precondition:	The user needs to be logged in
Postcondition:	The transaction has been booked in the system and the user has been notified about the success
Additional users:	None
Standard procedure:	1. The user navigates to the desired portfolio view 2. The system displays an overview of the desired portfolio with control elements for purchase and sale 3. The user clicks on the control element for the desired transaction type 4. The system displays a form with fields to select stock and amount 5. The user selects a stock and sets the amount 6. The system calculates the transaction value and displays it in the form 7. The user confirms the input 8. The system validates the user input 9. The input does not contain any errors 10. The system sets an unique transaction number and timestamp, books the transaction and shows the user a success notification
Extensions:	1. – 8. Like standard procedure 9. The input contains errors 10. The system shows the user an error notification

3 Generating Use Case Descriptions using ChatGPT

To explore ChatGPT's capabilities in generating use case descriptions, a structured approach was followed using three case studies. The ChatGPT used is based on free form of GPT-4 model. Here are the steps of the process:

3.1 Initial Case Study: ARENA

In order to create a systematic approach first trials has been done via a case study called ARENA. This is a well known case study presented in a software engineering textbook by Bruegge and Dutoit [5]. The initial prompt to ChatGPT is as follows:

> "I want to generate several use case descriptions for my project based on the details below. Provide use case descriptions tables in the format of containing rows Use Case Name, Primary Actors, Goal, Precondition, Postcondition at success and Standard scenario. Make it detailed and expansive and feel free to add your own ideas to it as well."

After that ARENA case study description text was fed to ChatGPT. The text is composed of the description and a dedicated section of functional requirements. As a result, ChatGPT provided a big table with 10 use case descriptions, primarily covering functional requirements, but not all requirements were addressed.

3.2 Refining the Prompt

The prompt was then modified to generate each use case description separately and adding the guidelines mentioned in Section 4.1.1:

> "Generate each use case description separately and follow these guidelines while writing."

ChatGPT generated same 10 use case descriptions without any affect on the text but on separate tables. Missing requirements were still not addressed and the guidelines were not followed.

3.3 Ensuring Comprehensive Coverage

To ensure comprehensive coverage, the prompt was further refined by adding:

> "Make sure that every functional requirement for each actor is covered in a separate use case description."

This adjustment led to the generation of two additional use case descriptions, but still not all requirements were covered.

3.4 Second Case Study: Surf and Sailing Center

In order to see ChatGPT's capabilities on another case study, we selected the case study "Surf and Sailing Center". This case study is part of our teaching material of the software engineering course taught at Bielefeld University of Applied Sciences. To the best of our knowledge, the case study is not included in any published textbook and is not publicly available. We deliberately selected this case study in order to avoid the situation that ChatGPT might have been trained with this material.

This case study, unlike the ARENA case study, does not have a dedicated section for functional requirements but contains a more general description of the system to be build. The same prompt as in the ARENA case study is used.

ChatGPT generated four use case descriptions separately and indicated that more could be generated if needed. Not having explicit functional requirements in the case study had no effect other than providing less use case description in the initial prompt. Although there was no clear comment on creating each use case definition separately, the use case descriptions were generated in separate tables.

3.5 Requesting More Use Cases

To generate more use case descriptions the follow-up prompt "need more" was repeatedly given. ChatGPT generated three to four additional descriptions each time. This iterative process continued until 20 use case descriptions was generated.

3.6 Final Case Study: Stock Market

Based on this experience, we then selected the stock market case study introduced earlier. This case study is solely defined for our teaching material. To the best of our knowledge, this case study is not publicly available, like surf and sailing center case study described in Section 3.4. Over the past years, it has been used in various student projects in the context where students were required to create use case descriptions and then design and implement a web-based application. This study was chosen to compare ChatGPT-generated and human-written use case descriptions.

The initial prompt used was the same as in ARENA case study. ChatGPT responded as follows:

> "These are the initial use case descriptions. Let me know if you need more use cases or any modifications!"

the prompt "need more" was repeatedly given until 16 use case descriptions were generated. 16 was chosen as it is the median of the two student sets having the highest number of use case descriptions i.e. 11 and 21 (see Table 3). The set contained a duplicate use case description for registering a user, named as "User Registration" and "Register User". Both of them have been included in the materials.

An example use case description generated by ChatGPT can be seen in Table 2.

Table 2: ChatGPT-generated "Perform Stock Transaction" use case description from stock market case study.

Use Case Name	**Perform Stock Transaction**
Primary Actors	User
Goal	To perform a transaction for a stock
Precondition	User is logged in, Depot exists
Postcondition	Transaction is recorded
Standard Scenario	1. User selects the depot in which they want to perform the transaction. 2. User navigates to the "Transactions" section. 3. User selects the type of transaction (buy or sell). 4. User enters the necessary information such as stock, quantity, transaction value, etc. 5. User submits the transaction form. 6. System validates the entered information. 7. If validation is successful and the transaction is feasible, the system records the transaction. 8. If validation fails or the transaction is not feasible, an error message is displayed.

This iterative process aimed the generation of use case descriptions by ChatGPT. The insights gained might shed light for assessing ChatGPT's potential in assisting use case description generation in software engineering.

4 Evaluating ChatGPT's capabilities

Generating use case descriptions by ChatGPT raises the question of what quality the use case descriptions are and how they compare to use case descriptions written by humans. In order to address this question, we have conducted an observational study.

The study includes the use case descriptions generated by ChatGPT and use case descriptions taken from three student projects using the stock market case study (see Section 2.2). The quality assessment is done via a survey including questions about the quality of use case descriptions and presence of bad smells.

In the following sections, the survey questions consisting of the quality criteria and bad smells of use case descriptions will be presented. After that the participants of the study and the data collection details will be provided.

4.1 Materials

Four sets of use case descriptions have been used in the study. Three sets of the use case descriptions are taken from the project report of the students who submit the project at their second or third year of the studies. They have all consented to share their materials to be used in this study. All personal information has been excluded from the materials. The other set of use case descriptions are generated by ChatGPT following the steps described in Section 3.

There are in total 58 use case descriptions. The number of use case descriptions from each use case description solution set can be found in Table 3. The solution sets from students projects are named as GP6, GP7 and GP24, and the ones generated by ChatGPT are named as CT.

Table 3: Use case description solution set names & the number of use case descriptions in each set

Solution Set Name/ Abbreviation	Number Of use Case Descriptions
ChatGPT / CT	16
Group 6 / GP6	11
Group 7 / GP7	10
Group 24 / GP24	21

4.1.1 Survey. A survey is used to evaluate the quality and presence of bad smells in the use case descriptions. The survey consists of 13 questions and it is presented in two pages. It takes about eight minutes to complete. Seven questions are from quality criteria and six questions are from bad smells. A short description of the study and a link to the stock market case study description are provided at the beginning of the survey. A use case description is provided as an image at the beginning of both pages. The first page is dedicated to quality criteria and second page is dedicated to bad smells. A short description provided in the beginning of the survey as follows:

"Welcome to use case description quality and bad smells survey! The survey is composed of 13 question and takes about 8 minutes.

The following use case description is from stock market use case in the SoSe exercises. The description can be found here (see Section 2.2). Based on this stock market use case, please examine the following use case description and answer the following questions."

The quality criteria and bad smells questions are covered separately in the following sections.

Quality Criteria. Since there is no commonly accepted quality criteria for use case descriptions, the quality criteria statements have been gathered from [4] and [13]. The statements that are easily explainable to students and easily applicable to the use case descriptions were selected. After that the statements were put in a form to be rated in a Likert scale in the survey. The statements can be seen in Table 4:

Table 4: Quality criteria statements

	Quality Criteria Statements
1	Use cases are named with verbs.
2	Actors are named with nouns.
3	Use case does not exceed two pages.
4	Use simple sentence structures
5	Use a neutral description (from a "bird's eye view")
6	Use active sentences
7	Clearly state the causal connection between successive steps
8	Clearly distinguish user steps from the step carried out by the
9	Represent the entire user transaction
10	Refer to the function, DO NOT describe the UI

Since first three statements are either right or wrong, they are excluded from the survey and answered only once with one participant. The rest of the seven statements have been tailored to be rated as Likert scale statements (scaled from "Strongly disagree" to "Strongly agree") and some examples and explanations have been provided to make it more clear to the participants. The tailored statements can be found in the Table 6.

Bad Smells. Bad smells are gathered from [15] and some explanations are added. In addition, at the beginning of the section, a sample use case description from our lecture containing bad smells has been presented in an image form and each bad smell is highlighted and discussed to give a better understanding to the participants and gather more useful answers from them.

The participants are asked to rate the existence of bad smells in the use case description on a Likert scale starting from "Not at all" to "Very much". The statements can be seen in Table 5.

4.2 Participants

The participants of the study are second and third year students of the Department of Business Informatics at Bielefeld University of Applied Sciences in Germany. They have been introduced the concepts of quality criteria and bad smells of use case descriptions in the System Development - Software Design course, in English,

Table 5: Bad smells questions of use case descriptions

	Please rate the existence of bad smells in the use case description below:
1.	Ambiguity: unclear or confusing, or it can be understood in more than one way
2.	Incorrectness: wrong or inconsistent content
3.	Poor Granularity: too rough or imprecise description
4.	Redundancy: unnecessary steps in the description
5.	Missing information : necessary information is missing
6.	Misplacement: description content in the wrong place

with the materials presented in Section 4.1.1. Stock Market case study is one of the case studies that have been used in this course's exercises. Therefore, they are suitable population to participate in the surveys. The participants consist of 23 German speaking students, five female and 18 male.

4.3 Data Collection

In the survey, use case descriptions are provided in German language, while the quality criteria and bad smells statements to be rated are presented in English (as they have been introduced in English in the lecture). The use case descriptions from student projects were already in German. The use case descriptions from ChatGPT have been translated to German. They were proofread only for language specific corrections and the content has not been altered.

A naming convention consisting of the solution set name (explained in Section 4.1), use case description number, and use case description name is used to distinguish use case descriptions from each other. An example use case description name from ChatGPT solution set with the name "Register User" is as follows: "STM-CT-01-RegisterUser."

Since each project consists of different number of use case descriptions (see Table 3), they have been divided almost equally into 12 groups. Each group contains either four, five or six surveys. Each group contains use case descriptions from one solution set, so that the participants are not surveying overlapping use case descriptions from different solution sets, and also they are exposed to only one solution set's writing style.

The participants have been informed the purpose of the study, and they volunteered to participate. There were no monetary or grade contribution to the participants. 23 participants are randomly assigned to conduct the survey from one of the predefined 12 groups.

It took 14 days to complete the surveys. As soon as the surveys have been completed the participants have been anonymized and all the personal information have been deleted.

5 Results

A total of 23 participants conducted the survey during the first 14 days. The results from the quality criteria and bad smells of the use case descriptions are analyzed separately in the following subsections.

5.1 Quality Criteria

There were in total seven quality criteria questions in the survey. These questions are rated on a scale from "Strongly disagree" to "Strongly agree", and they are converted into scores ranging from one to five by Google Forms. Since the questions are asking quality such as: "Simple sentence structures are used" (see Table 6 for all the questions), higher scores indicate better quality use case descriptions. In this study the use case descriptions with scores from one to 2.5 will be treated as having "low quality", from 2.5 to 3.5 will be treated as having "medium quality", and the scores above 3.5 will be treated as having "high quality".

Table 6: Quality criteria questions of use case descriptions

1.	Simple sentence structures are used. **Ex**: noun ... verb ... direct Object . . . prepositional phrase
2.	A neutral description (from a "bird's eye view") is used. **Ex**: " The system . . . posts . . . the sum . . . to the current account
3.	Active sentences are used, it is clearly stated who carries out the step **Ex**: "The system … posts … the sum … to the current account"
4.	The causal connection between successive steps is clearly stated.
5.	The steps carried out by the system are clearly distinguished from the steps carried out by users.
6.	Entire user transaction is covered. (All the steps from beginning to end are described)
7.	The function to be executed is referred to instead of describing the user interface. **Ex**: "… activates the "Create Customer" function" NOT "… clicks the "Create Customer" button"

For the analysis, the responses have been averaged per solution set from the sets: CT (ChatGPT), Group 6 (GP6), Group 7 (GP7) and Group 24 (GP24) (see section 4.1 for more information). The average quality scores per solution set can be seen in Table 7. The visual representation of average quality scores can be seen in Figure 2. It can be seen from the table and the figure that the GP6 has medium quality use case descriptions, while other three projects have high quality use case descriptions. It can be observed that ChatGPT has performed very well and compares well to human written use case descriptions that have high quality. More specifically, it is the second best solution set with high quality. The statistical significance is planned to be computed in the future while covering more case studies.

5.2 Bad Smells

There were in total six bad smell questions in the survey. These questions are rated on a scale from "Not at all" to "Very much", and they are converted into scores ranging from one to five by Google Forms. Since the questions are asking existence of bad smells such as existence of "Ambiguity" (see Table 5 for all the bad smells), lower scores are indicating a use case description with less bad smells. In this study the use case descriptions with scores from one to 2.5 will

Table 7: Average Quality Criteria Scores of Solution Sets

Solution Set Name	Average Quality Criteria Scores
CT	4.05
GP6	3.21
GP7	4.06
GP24	3.91

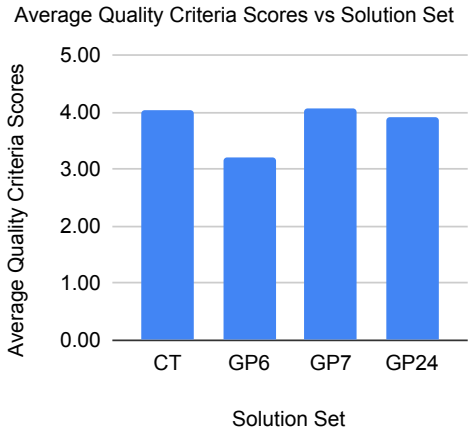

Figure 2: Average quality criteria scores versus solution set names

be treated as having "less bad smells", from 2.5 to 3.5 will be treated as having "medium bad smells", and the scores above 3.5 will be treated as having "lots of bad smells".

As it can be seen from Table 8 and Figure 3 CT (ChatGPT) has overall the lowest bad smells in the use case descriptions followed by GP7 and GP24. These sets are categorized as having "less bad smells". GP6 is categorized as having "medium bad smells".

Table 8: Average Bad Smell Scores of Solution Sets

Solution Set Name	Average Bad Smell Scores
CT	2.16
GP6	3.04
GP7	2.18
GP24	2.27

Specifically, it can be seen from Figure 4 that ChatGPT has the least bad smells in the categories of Ambiguity, Incorrectness and Misplacement, while it has a bit more bad smells from human written use case descriptions in the poor granularity, redundancy and missing information categories.

6 Conclusion

Generating use case descriptions with ChatGPT gives rise to the problem of assessing the quality of use case descriptions generated.

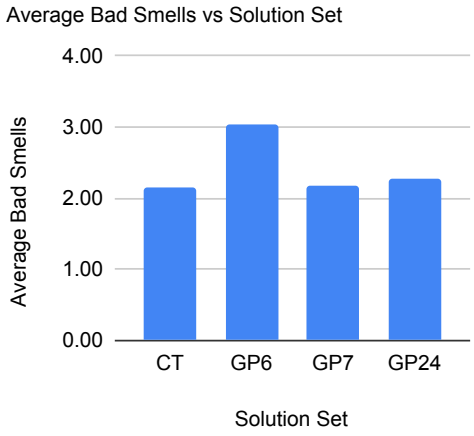

Figure 3: Average bad smells existence scores in the solution sets

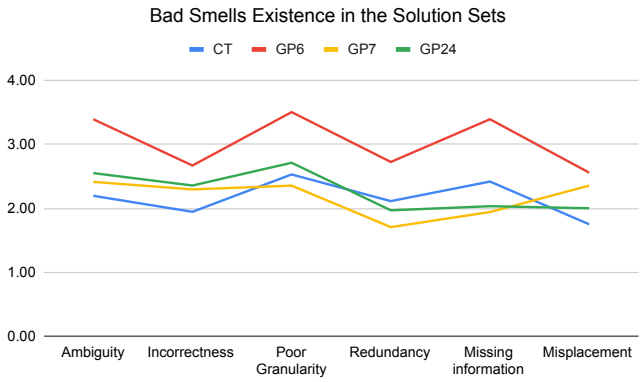

Figure 4: Existence of bad smells in the use case descriptions in solution sets

In particular with regard to human written use case descriptions. In this paper we have presented an approach to systematically evaluate the quality of use case descriptions. Part of our approach is to do a comparative analysis.

In this comparative analysis of ChatGPT-generated and human-written use case descriptions, we sought to determine the quality of AI-generated content within the domain of software engineering. Our study revealed that ChatGPT possesses an ability to produce high quality use case descriptions having less bad smells in the ambiguity, incorrectness and misplacement categories and a little bit more bad smells in the categories of poor granularity, redundancy and missing information. The results demonstrate the potential for AI to assist in the creation of functional requirements documentation in terms of use case descriptions.

However ChatGPT can generate duplicated use case descriptions with slightly different naming (such as "User Registration" and "Register User"). In these cases human experts should eliminate one

of them considering the quality criteria and bad smells presented in Sections 4.1.1 and 4.1.1. In this study "User Registration" should be the one to be eliminated as quality criteria suggest that use case descriptions should be named with verbs.

Despite some limitations, ChatGPT's performance suggests that AI tools might serve as an aid in the software development process. They might enhance efficiency by generating initial drafts of use case descriptions, which can then be refined by human experts. This hybrid approach leverages the strengths of both AI and human intelligence, potentially improving the overall productivity and accuracy of requirements documentation.

7 Future Work

Building on the findings of this comparative analysis of ChatGPT-generated and human-written use case descriptions, several avenues for future research can enhance the robustness and applicability of the conclusions drawn in this study. Incorporating different case studies across various domains and complexities can help generalize the findings. Future research should include a larger and more diverse group of participants, encompassing a wide range of expertise levels and industry backgrounds. Expanding the scope of the analysis is another direction to cover a broader range of "bad smells" that will provide a more detailed assessment of ChatGPT's performance. Conducting longitudinal studies to assess the long-term impact of using AI-generated use case descriptions on project success and development efficiency would provide a deeper understanding of their practical value.

By addressing these areas in future research, the field can advance towards more effective integration of AI in software engineering. For instance ChatGPT generated use case descriptions could be integrated into existing MDE workflows. Thereby the quality and efficiency of requirements documentation and the software development process as a whole could be improved.

Acknowledgments

We would like to express our sincere gratitude to Felix Lennart Schildmann for his invaluable contributions to this article. We would also like to thank all the students who shared their project solution sets with us and all the participants who volunteered to fill out the survey.

References

[1] Camille Achour, Colette Rolland, Carine Souveyet, and Neil Maiden. 1999. Guiding Use Case Authoring: Results of an Empirical Study. *International Symposium on Requirements Engineering (ISRE)*, 36–43. https://doi.org/10.1109/ISRE.1999.777983

[2] Steve Adolph, Paul Bramble, Alistair Cockburn, and Andy Pols. 2002. *Patterns for Effective Use Cases.* Pearson Education.

[3] Leila Bencheikh and Niklas Höglund. 2023. Exploring the efficacy of chatgpt in generating requirements: An experimental study. *Bachelor's Thesis, Chalmers University of Technology, Göteborg, Sweden* (2023).

[4] Bernd Bruegge and Allen H. Dutoit. 2009. *Object-Oriented Software Engineering Using UML, Patterns, and Java* (3rd ed.). Prentice Hall Press, USA.

[5] Bernd Bruegge and Allen .H. Dutoit. 2014. *Object-oriented software engineering using UML, patterns, and Java.* Pearson. https://www.pearson.de/object-oriented-software-engineering-using-uml-patterns-and-java-pearson-new-international-edition-9781292024011

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[7] Alistair Cockburn. 1998. Basic use case template. *Humans and Technology, Technical Report* 96 (1998), 28.

[8] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. *arXiv preprint arXiv:2310.03533* (2023).

[9] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, and Gunnar Övergaard. 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley, Reading.

[10] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. 2022. Competition-level code generation with alphacode. *Science* 378, 6624 (2022), 1092–1097.

[11] Agnieszka Marczak-Czajka and Jane Cleland-Huang. 2023. Using chatgpt to generate human-value user stories as inspirational triggers. In *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW).* IEEE, 52–61.

[12] Nuno Marques, Rodrigo Rocha Silva, and Jorge Bernardino. 2024. Using ChatGPT in Software Requirements Engineering: A Comprehensive Review. *Future Internet* 16, 6 (2024), 180.

[13] Keith Thomas Phalp, Jonathan Vincent, and Karl Cox. 2007. Assessing the quality of use case descriptions. *Software Quality Journal* 15, 1 (mar 2007), 69–97. https://doi.org/10.1007/s11219-006-9006-z

[14] Krishna Ronanki, Beatriz Cabrero-Daniel, and Christian Berger. 2022. ChatGPT as a Tool for User Story Quality Evaluation: Trustworthy Out of the Box?. In *International Conference on Agile Software Development.* Springer, 173–181.

[15] Yotaro Seki, Shinpei Hayashi, and Motoshi Saeki. 2019. Detecting Bad Smells in Use Case Descriptions. In *2019 IEEE 27th International Requirements Engineering Conference (RE).* IEEE. https://doi.org/10.1109/re.2019.00021

[16] Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, and Qing Wang. 2024. Software testing with large language models: Survey, landscape, and vision. *IEEE Transactions on Software Engineering* (2024).

[17] Burak Yetiştiren, Işık Özsoy, Miray Ayerdem, and Eray Tüzün. 2023. Evaluating the code quality of ai-assisted code generation tools: An empirical study on github copilot, amazon codewhisperer, and chatgpt. *arXiv preprint arXiv:2304.10778* (2023).

[18] Yuanyuan Zhang, Anthony Finkelstein, and Mark Harman. 2008. Search based requirements optimisation: Existing work and challenges. In *Requirements Engineering: Foundation for Software Quality: 14th International Working Conference, REFSQ 2008 Montpellier, France, June 16-17, 2008 Proceedings 14.* Springer, 88–94.