Berger Scientfic Press Inc.
Innovation & Technology Advances
Volume 1 Issue 1, June, 2023
www.bergersci.com
DOI: https://doi.org/10.61187/ita.v1i1.19

An Empirical Study of Adoption of ChatGPT for Bug Fixing among Professional Developers

Haotong Ge, Yuemeng Wu

Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada
haotong.ge@mail.utoronto.ca
yuemeng.wu@mail.utoronto.ca

Abstract: ***ChatGPT is a powerful tool that assists software engineers in identifying and rectifying errors in code. One of its primary advantages is its ability to engage in natural language conversation with humans, which allows it to collaborate more closely with engineers in improving and optimizing the code. However, despite its potential advantages, software developers do not always utilize ChatGPT as a tool for bug fixing. In this study, we aim to examine the factors that influence the adoption of ChatGPT for bug fixing among professional software developers, based on the Unified Theory of Acceptance and Use of Technology (UTAUT) theory. To accomplish this, we conducted 50 semi-structured interviews with professional software developers and other stakeholders. Our findings indicate that the performance expectancy and effort expectancy of professional software developers, as well as social influence, facilitating conditions, data security, and trust are the key factors of adoption. These findings suggest that understanding these factors can be critical in promoting the adoption and use of ChatGPT in the software development industry.***

Keywords: software engineering, adoption, social factors, ChatGPT.

1. Introduction

Debugging is a common, yet time-consuming and challenging aspect of the software engineering process [1]. It is widely accepted that the debugging process is usually imperfect and that the impact of imperfect debugging on software development costs is significant, which in turn may affect the optimal software release time or operating budget [2]. One of the main issues in software development is the cost problem. It is well-known that the development of a software system can be very time-consuming and expensive. For example, the cost of debugging, testing, and verification is estimated to account for 50-75 percent of the total budget of software development projects amounting to more than 10 billion dollars annually [3]. Complex software usually contains undiscovered bugs in its source code. The later these bugs are discovered, the more severe and extensive their consequences can be. Uncorrected bugs in software can lead to failures of essential systems, which can result in high economic costs. However, the process of finding and fixing bugs in software requires significant skills, knowledge, and resources. This has led researchers to explore ways to help developers fix bugs more efficiently.

Open Artificial Intelligence (AI) released an AI chat tool called ChatGPT in late November 2022. ChatGPT has quickly gained popularity on the internet. This chatbot is built on top of OpenAI's language model [4] and allows users to communicate with the AI through prompt-based conversations. Users can ask questions to the bot, and ChatGPT will respond with pertinent and persuasive topics and responses.

ChatGPT's unique ability to communicate with humans makes it a valuable tool for assisting software engineers in identifying and fixing errors in computer code. With advanced NLP capabilities, ChatGPT can understand the intent behind the code and identify potential errors based on the language used. Additionally, its large-scale training on text data enables it to suggest fixes to the code, thus reducing time and effort [5].

Due to the nature of the interaction mode between ChatGPT and users, our focus of bug fixing is on a specific scenario where users can input error messages and existing code, and then ask ChatGPT for suggestions on how to fix the errors. To illustrate this scenario, Figure 1 presents a simple example of ChatGPT assisting with bug fixing by providing suggestions based on an error message and corresponding code.

Traditionally, developers may need to search for similar issues that have been previously encountered and resolved by other developers. This allows for a quick and efficient resolution of the bug without having to spend time and resources trying to troubleshoot the issue from scratch. However, the answers may not always be tailored to the user's specific issue. With ChatGPT's ability to provide personalized and contextualized assistance, it can be a faster and more efficient way to get the job done than searching through complex and often lengthy answers.

Moreover, recent research by Sobania et al. [6] has demonstrated that ChatGPT can effectively fix code better than existing programs. The error repair performance of ChatGPT is competitive with the common deep learning methods, such as CoCoNut and Codex, and significantly outperforms the reported results of standard program repair methods. In contrast to previous approaches, ChatGPT provides a dialogue system through which further information can be input, for example, the expected output of a given input or observed error information. In fact, by providing ChatGPT with such hints, its success rate can be further improved, fixing 31 out of 40 errors, exceeding the state-of-the-art.

Despite ChatGPT's competitive performance among already existing debugging software, we found that ChatGPT is not very commonly adopted by software developers for bug fixing. Therefore, we focus on whether ChatGPT would be adopted by professional developers and the factors that would influence its adoption.

Our objective is to study the utilization of ChatGPT among professional developers. For the purposes of our study, we define professional developers as individuals who are 18 years of age or older, employed in a software development-related job, and have written code as part of their professional activities. To gather various perspectives on the use of ChatGPT, we will also recruit other stakeholders such as project managers and security engineers as participants.

In this paper, we examine the adoption of ChatGPT for bug fixing by studying the factors that affect the adoption of ChatGPT. Our study is based on the Unified Theory of Acceptance and Use of Technology (UTAUT) [7], which offers a theoretical framework for comprehending the factors that shape people's acceptance and utilization of novel technology.

The main contribution of this paper is a qualitative analysis of the factors that influence the adoption of ChatGPT for bug fixing and how these factors interact with each other. Our study is based on interviews with 50 professional software developers and other stakeholders that may affect the adoption of ChatGPT such as project managers, quality assurance engineers, and security engineers. The results of the study can help developers of ChatGPT to improve the technology, and can also provide researchers with insights into additional potential factors that may affect its adoption.

Here, we set out to explore the following research questions:

- Will professional developers adopt ChatGPT, a new technology, for bug fixing?
- What are the factors that influence professional developers' decisions to adopt or reject ChatGPT for bug fixing?

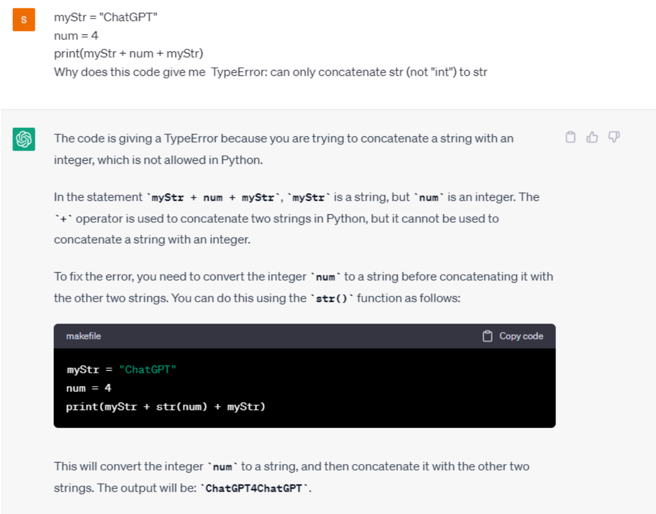

Figure 1: Simple example of bug fixing using ChatGPT.

2. Related work

Our research draws upon established social science theories of adoption and applies them to the context of software engineering. In this section, we provide a review of existing studies on the adoption of software engineering technology, as well as an introduction of the grounded theory that underpins our own research.

2.1 Adoption of Software Engineering Technologies

As new technologies continue to emerge, researchers have become increasingly interested in understanding how these innovations can be adopted by individuals and organizations. In the field of software engineering, the adoption of new technologies is of particular importance, as it can significantly affect a team's ability to develop software efficiently and effectively.

Understanding the factors that influence the adoption of new technologies in software engineering is a critical area of research. Dybaa et al. [8] conducted a systematic review of existing research on the adoption of agile development as a software engineering methodology. Their study aimed to provide a comprehensive overview of the benefits and limitations of agile methods, as well as the strength of evidence supporting their use in software development. Kochhar et al. [9] examined a large number of non-trivial software projects and investigated the relationship between test cases and various project development characteristics, in order to gain a better understanding of the adoption of software testing in open-source projects. Narrowing down to specific software, Awa et al. [10] focused on identifying the critical adoption factors of Enterprise Resource Planning (ERP) software. To explore the adoption factors, they conducted a survey and determined that technological, organizational, and environmental exogenous factors were significant determinants of adoption.

To develop a more coherent understanding of adoption factors, researchers have increasingly begun to incorporate established social science theories into their studies. By framing their research in this way, they are able to provide a stronger theoretical foundation for their findings and leverage existing theories to bolster their conclusions.

One example of an established social science theory that has been applied to the field of software engineering is the diffusion of innovations (DOI) [11]. This theory has been used to inspire studies on the adoption of software engineering technologies. For example, Xiao et al. [12] conducted a study based on the DOI theory, which aimed to investigate the impact of social factors on the adoption of security tools within software development companies. Similarly, Lambda et al. [13] utilized the DOI theory to form hypotheses on the factors that influence the adoption of automation tools in open-source communities and analyzed a large longitudinal dataset to verify their hypotheses.

In addition to the DOI theory, the Technology Acceptance Model (TAM) is also widely used in studies of technology adoption. According to TAM, perceived usefulness and perceived ease of use are two key factors that influence individuals' acceptance of a technology. An example of applying TAM to software engineering practices is the study conducted by Wallance and Sheetz [14]. Specifically, their study focused on the adoption of software measures for project management and software process improvement. By examining the perceived usefulness and perceived ease of use of these measures, the study provided guidance for software engineers in selecting the most suitable software measures.

2.2 Unified Theory of Acceptance and Use of Technology

In our study, we aim to investigate the factors that affect the adoption of ChatGPT for bug fixing among professional developers. In order to ensure that our study is comprehensive and well-founded, we have chosen to use an established theory, the Unified Theory of Acceptance and Use of Technology (UTAUT) [7], to guide our investigation.

The UTAUT framework has been widely used to explain the adoption of new technologies within organizations. It synthesizes elements from eight prominent models to provide insights into the factors that affect the adoption of new technologies, including theory of reasoned action (TRA) [15], TAM, motivation model (MM) [16], theory of planned behavior (TPB) [17], combined TAM and TPB (C-TAM-TPB) [18], model of PC utilization (MPCU) [19], DOI, and social cognitive theory (SCT) [20]. Therefore, we choose UTAUT as the theoretical foundation for our study, as it can provide a more comprehensive understanding of the factors that may influence the adoption of ChatGPT for bug fixing.

The UTAUT theory identified four core determinants that affect the acceptance and usage of technology: performance expectancy, effort expectancy, social influence, and facilitating conditions. Performance expectancy refers to the degree to which an individual believes that using a particular system will improve their job performance, while effort expectancy relates to the ease of use of the system. Social influence is the extent to which an individual perceives that others think they should use the new system, and facilitating conditions refer to the level of support available for using the system.

In addition to the four core determinants, the UTAUT theory identifies four key moderators that may influence the relationship between the determinants and the adoption of new technology. These moderators include gender, age, voluntariness, and experience. Gender and age may impact the adoption of new technology due to differences in attitudes and beliefs between different demographic groups. Voluntariness refers to the extent to which an individual has a choice in using the new technology, as those who are forced to use it may have different attitudes and behaviors than those who choose to use it. Experience refers to an individual's previous experience with similar technologies, which may affect their perceptions of the new technology and its usefulness. The UTAUT theory acknowledges that these moderators may dilute the impact of the four determinants and must be taken into consideration when studying the adoption of new technology.

According to the UTAUT theory, the interplay between the four determinants and four moderators can be illustrated through Figure 2. This figure shows that the four determinants have a direct effect on the intention to use the new technology. Additionally, the four moderators can influence the strength of the relationship between the determinants and the intention to use the technology, either positively or negatively. For instance, the influence of performance expectancy is moderated by gender and age, with younger men having a stronger effect. The influence of effort expectancy is moderated by gender, age, and experience, with younger women in early stages of experience having a stronger effect. The influence of social influence is moderated by gender, age, voluntariness, and experience, with older women in early stages of experience and mandatory settings having a stronger effect. Finally, the influence of facilitating conditions is moderated by age and experience, with older and more experienced people having a stronger effect.

To our knowledge, there has been little investigation on the use of ChatGPT for bug fixing in the context of professional software development, and none of them have specifically used the UTAUT framework as the theoretical foundation. Therefore, we plan to use the UTAUT theory to identify and analyze the key factors that could influence the intention of professional developers to adopt ChatGPT for bug fixing. By doing so, we aim to contribute to a better understanding of the potential barriers and drivers of adoption of ChatGPT in the software development industry, and provide insights for the design and implementation of more effective tools and practices that can be widely accepted by the users.

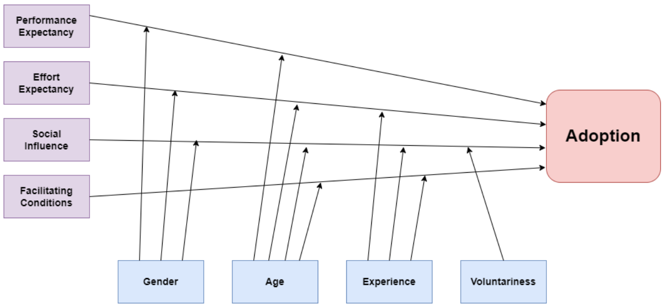

Figure 2: The relationship of the four determinants and four moderators in UTAUT.

3. Methodology

We plan to conduct semi-structured interviews to investigate the factors that influence the adoption of ChatGPT among professional software developers. Specifically, we aim to understand their intentions to adopt or reject ChatGPT and the rationales behind their decisions.

Our research will qualitatively describe the factors that may influence the adoption of ChatGPT and how these factors interact with each other.

3.1 Participant Recruitment

We plan to recruit participants through various means. Firstly, we will search for potential participants among open source projects on GitHub and utilize discussion channels to reach out to them. Secondly, we will post recruitment information on online developer communities and forums, including Reddit's programming subreddits and Hacker News. Thirdly, we will explore recruitment websites such as User Interviews that connect researchers with participants. Finally, we will also ask recruited participants to recommend their colleagues or friends who meet our criteria.

Once the recruitment process is complete, we will administer a screening survey to ensure that the participants meet our criteria of being professional software developers or other stakeholders who may affect the adoption of ChatGPT, such as project managers.

We plan to recruit participants from companies of different sizes and areas in order to provide more generalizability to our study. Our recruitment strategy is designed to attract developers from companies of various sizes, including large, medium, and small organizations. Additionally, we will be inviting professionals in various roles, including project managers, security developers, and quality assurance engineers to participate in our project. We also aim to have a diverse sample that includes both female and male participants, as well as individuals from different age groups. By involving individuals with diverse backgrounds and skill sets, we can achieve a more comprehensive result.

We anticipate that our sample size will be 50 participants, as conducting semi-structured interviews is time-intensive. After analyzing the data from these interviews, we plan to conduct a survey to include more participants, which will help us to draw more generalizable conclusions in our future research.

Table 1 presents the expected number of participants, categorized by development roles and the size of the companies they work for. We have classified companies into three groups: small, medium, and large, based on the number of developers they employ. Small companies have 100 or fewer developers, medium companies have more than 100 but fewer than 1000 developers, while large companies employ 1000 or more developers.

As shown in the table, we intend to conduct interviews with 36 programmers, 6 project managers, 4 security engineers, and 4 quality assurance engineers. The majority of our participants are programmers, as we are particularly interested in their experience with ChatGPT for bug fixing. However, we also plan to interview other stakeholders to ensure a more comprehensive result. The table also indicates the number of participants recruited from companies of different sizes. We have planned to interview with 10 participants from small companies, 17 from medium-sized companies, and 23 from large companies.

We will not specify the number of female or male participants or the number of participants from different age groups in advance. Instead, we will control the number during our recruitment process. Our goal is to have a relatively balanced representation of participants across different age groups and genders.

Table 1: Distribution of developers based on company size.

Company Size	Programmer	Manager	Security Engineers	Quality Assurance Engineers	Total
small	7	1	1	1	10
medium	13	2	1	1	17
large	16	3	2	2	23
total	36	6	4	4	50

3.2 Interview Methodology

Semi-structured interviews [21] are particularly useful for exploratory studies, as they offer greater flexibility in coverage and tend to generate richer data compared to other interview styles [22]. Given our aim of gaining new insights into the factors that drive professional developers to adopt ChatGPT for bug fixing, we believe that semi-structured interviews are the most suitable approach for our study.

We will conduct our semi-structured interviews through video conferencing software, such as Zoom or Skype, and record the interviews with the participants' consent. Each interview will last approximately 30-45 minutes. After conducting the interviews, we will transcribe and analyze the data to gain insights into the factors that drive professional developers to adopt or reject ChatGPT for bug fixing.

Before conducting an interview for our study, we will first obtain permission from the participant to record the conversation and use their responses for research purposes. Additionally, if the participant is not familiar with ChatGPT, we will provide a brief introduction and demonstrate its functionality.

We will use an interview script to guide our interview. To design our interview script, we will use the UTAUT theory. As we mentioned earlier, the UTAUT theory proposes four main factors that affect technology acceptance and usage: performance expectancy, effort expectancy, social influence, and facilitating conditions. Therefore, our interview questions will be designed to explore these factors and how they relate to professional developers' decision to adopt or reject ChatGPT for bug fixing. We will also consider the four key moderators: gender, age, voluntariness, and experience when designing our script.

Our interview script includes both open-ended questions about participants' adoption choices of ChatGPT, as well as more focused questions based on the factors of the UTAUT theory. For instance, to examine performance expectancy, we may inquire, "Do you expect ChatGPT to enhance your productivity when fixing bugs?". A comprehensive version of our interview questions can be found in Appendix A.

During the interview process, our interviewers have the flexibility to ask additional questions if they think a participant's response is particularly interesting. These questions help us gather more comprehensive data and may offer new insights to our research. If we discover that a follow-up question yields valuable information, we will add it to our script for subsequent interviews. Conversely, if we find that a question is no longer providing new insights, we will discontinue asking it to that particular participant.

If time permits, we will also inquire with managers, security experts, and other stakeholders about their activities beyond programming. Specifically, we ask managers about the factors they consider when deciding whether to recommend using ChatGPT for bug fixing in their team; we ask security experts whether there are potential security concerns regarding the adoption of ChatGPT and whether they believe such concerns could hinder user adoption.

After the interview, we will thank the participant for their time and reiterate the purpose of our study. We will also offer the opportunity for the participant to review the transcript of the interview and make any necessary corrections. All data collected during the interview process will be kept confidential and anonymous.

3.3 Data Collection and Analysis

We will record the interviews we conducted and transcribe the audio into scripts for data analysis.

We then use NVivo to help organize and analyze the interview transcripts using open coding and axial coding. NVivo is a software program that allows researchers to code and categorize data, identify themes and patterns, and explore relationships between different pieces of information.

Our first step in analyzing the data is to use open coding to break it down into smaller segments and label each part as a code. We begin by establishing an initial set of codes based on the findings in UTAUT. As we analyze the data, we create additional codes to capture interesting patterns that were previously unidentified. Using open coding allows us to remain open to new theoretical possibilities from the data, rather than being constrained by preconceived ideas or theories. We then use axial coding to find connections between the codes and group codes together into categories. By doing so, we organize our codes into a more structured and systematic framework.

Our qualitative analysis aims to provide insights into how the adoption of ChatGPT is influenced by factors such as performance expectancy, effort expectancy, social influence, and facilitating conditions, along with key moderators such as gender, age, voluntariness, and experience. Additionally, the analysis may reveal new factors that impact the adoption of ChatGPT, such as data security.

Our work is exploratory and qualitative with a relatively small number of participants. Although our data analysis provides valuable insights into the factors that affect the adoption of ChatGPT in bug fixing for professional developers, we cannot make generalizations about developers' behavior based solely on our findings. However, our study can serve as a useful starting point for future research, which may help us achieve more generalized and robust results.

4. Expected Results

In this section, we will present the expected results of our study which analyzes the impact of four factors from the UTAUT theory, namely performance expectancy, effort expectancy, social influence, and facilitating conditions, along with the role of key moderators such as gender, age, voluntariness, and experience, which may moderate the effects of these factors. Our preliminary findings suggest that these four determinants have a significant influence on the decision of professional software developers to adopt ChatGPT for bug fixing. Moreover, we have also identified two novel factors that may impact the adoption of ChatGPT for this purpose, which will be discussed in detail in the following sections. Of the six factors we discovered, performance expectancy is the most important when making decisions about adopting ChatGPT.

4.1 Performance Expectancy

Performance expectancy is defined as the degree to which an individual believes that using the system will help him or her to attain gains in job performance [23]. In our study, several participants reported that their performance expectancy directly influenced their decision to use ChatGPT for bug fixing and their actual use of the tool.

More than half of the participants in our study reported that using ChatGPT for bug fixing at work would improve their job performance. Of the 24 participants who have already used ChatGPT for debugging, 13 reported that using the tool to fix code in their jobs would enable them to complete tasks more quickly and increase their productivity. This is because ChatGPT's ability to discuss can help them arrive at a more accurate answer, saving them time and effort. Additionally, nearly half of the participants mentioned that using ChatGPT could be useful for their job's code fixing requirements. Even those who have not used ChatGPT before still believe that this tool would be helpful for bug fixing and express willingness to try it out in the future. Therefore, the participants' performance expectancy regarding ChatGPT's impact on their job performance will strongly influence their decision to adopt the tool.

Furthermore, participants reported a general perception that using innovative tools is better than using their predecessors [7]. Specifically, 10 participants mentioned that ChatGPT would do a better job of fixing codes than the tool they were using before, leading to improved effectiveness and quality of work. As a new tool, ChatGPT's bug-fixing performance has attracted the attention of many participants who view it as superior to existing programs. Additionally, participants who have used ChatGPT to fix bugs reported its bug-fixing performance as competitive with common deep learning approaches such as CoCoNut and Codex, and significantly better than standard program repair approaches. Therefore, participants' performance expectancy regarding their attitude towards new tools will likely influence their decision to adopt ChatGPT.

The impact of performance expectancy on the adoption of ChatGPT for bug fixing will be moderated by age and gender. Studies have shown that men tend to be highly task-oriented, which means that performance expectancy focused on task completion has a greater impact on males [7]. Since bug fixing is a task with the goal of completing a task, performance expectancy will have a more prominent impact on men's adoption of ChatGPT for debugging. As a result, men are more likely to use ChatGPT to fix bugs at work, influenced by their performance expectancy. Additionally, younger software developers may be more focused on extrinsic rewards, and they will care more about their job performance. Therefore, younger people are more likely to be influenced by performance expectancy.

4.2 Effort Expectancy

Effort expectancy is defined as the level of ease associated with system usage. The indicators of effort expectancy are perceived ease of use and ease of use. Perceived ease of use is how far one believes that using a system will be free from difficult attempts. Ease of use is how far using innovation is perceived to be easy to use [23]. Our study finds that these factors have a significant impact on the decision to adopt ChatGPT for bug fixing.

There are many participants believe that using ChatGPT for bug fixing would be free of effort. Of the 44 professional software programmers, security and quality assurance engineers who participated in the interview, 28 participants reported that learning to use ChatGPT for bug fixing would be easy, and that it is easy to get ChatGPT to perform the tasks they need it to do. ChatGPT allows people to adapt to its use very quickly. It requires no training, and it responds to problems almost instantly. This is also true for bug fixing, as it will instantly find bugs and suggest fixes to software developers. Also, 25 participants mentioned that their interaction with ChatGPT would be clear and understandable since the advantage of ChatGPT is its unique ability to talk to humans directly. Therefore, perceived ease of use in effort expectancy can be considered an important factor in the adoption of ChatGPT for bug fixing.

In addition to perceived ease of use, software developers also consider the extent to which an innovation is considered difficult to use when deciding whether to adopt a tool for bug fixing. Of 24 participants who have already used ChatGPT to fix bugs, more than half of them reported that learning to operate ChatGPT for debugging was easy for them. They found ChatGPT very easy to get started in practice and would prefer to use ChatGPT to fix bugs in the future as a better alternative than the tools they have used to fix bugs before. However, 3 participants indicated that ChatGPT was not easy to use, as its ease of use did not meet their expectations, and they would not continue to use it in the future. Thus, based on the results of our interviews, we find that ChatGPT's actual ease of use in fixing bugs features had an impact on software developers' decisions to adopt it.

Moderators including age, gender, and experience have moderating effects on the relationship between effort expectancy and the adoption of ChatGPT to bug fixing. Since gender differences could be driven by cognitions related to gender roles [7], effort expectancy has a more significant impact on women than men. According to our respondents, when considering whether to adopt ChatGPT, female respondents are more concerned about its perceived ease of use and ease of use. Also, increased age is associated with difficulty in processing complex stimuli and allocating attention to information on the job [7], so older software developers are more likely to consider the ease of use of a tool for debugging. Our findings also confirm that constructs related to effort expectancy will be stronger determinants of the adoption of ChatGPT to fix code for women and older software developers. For example, women who are older and have relatively little experience in fixing bugs are more likely to focus on effort expectations as a factor in deciding whether to debug with ChatGPT.

4.3 Social Influence

The social factor is the internalization of a person's subjective culture to the reference group, and it is defined as the degree to which an individual perceives that important others believe he or she should use the new system [23]. In the context of our study, the important people who have a social influence on the respondents are primarily their colleagues and superiors, as all of our participants are software developers working in the industry.

Subjective norms are the main social factor influencing people's decision to adopt a tool, with most respondents indicating that the advice of those around them at work or the frequency of their use of ChatGPT influences software developers' decision to use ChatGPT to fix their code. 25 of 36 software programmers reported that they used ChatGPT because of the proportion of coworkers who use ChatGPT for bug fixing. Some of the respondents who have used ChatGPT to fix bugs say that part of the main reason they use this tool is that they have colleagues who use it to fix bugs and recommend it to them; others who had not used ChatGPT for bug fixing said they would be likely to try it if they had colleagues who use it and recommend it to fix code in work in the future. On the other hand, regulations imposed by superiors directly influenced the software developers' decision to use ChatGPT, with 21 participants reporting that their supervisors were supportive of using ChatGPT for debugging, which largely influenced their decision to use it. However, a small number of participants indicated that their superiors did not recommend using ChatGPT, so they continued their previous methods of fixing code.

The influence of social influence on the adoption of ChatGPT for bug fixing is moderated by gender, age, voluntariness, and experience. Based on the results of our study, we found that women tend to be more sensitive to the opinions of others. Therefore, in forming the intention to use ChatGPT, a new technology, we found that social factors have a more significant impact on women. Since affiliation needs to increase with age, older respondents are more likely to place increased salience on social influences. At the same time, the relationship between age and experience moderate social factors on the adoption decision of ChatGPT. Older and less experienced respondents indicated that they cared more about the advice of their colleagues as well as their supervisors. Overall, social influence has a strong impact on professional developers' decision to use ChatGPT for bug fixing, particularly for older and less experienced women.

4.4 Facilitating Conditions

Facilitating conditions are defined as the extent to which a person believes that the existing organizational and technical infrastructure supports the use of the system [23] and they are objective factors in the environment in which the observers agree to make an action to be easy to perform. Our results suggest that constructs including facilitating conditions and compatibility have a significant influence on the adoption of ChatGPT for bug fixing.

Favorable objective factors in the environment of ChatGPT make it easier for professional software developers to use ChatGPT for bug fixing. Among the 50 participants, 36 reported that they would be willing to adopt this tool if there was available guidance or a specific person to help them fix bugs using ChatGPT. Additionally, 33 respondents mentioned that using ChatGPT would be compatible with all aspects of their work in fixing bugs and fit into their work style. As the respondents are software development-related workers, they are exposed to work that is related to fixing code, and the use of ChatGPT is perceived as being consistent with their existing values and experiences, making it easier for them to adopt the tool.

Facilitating conditions are moderated by age and experience. Interviews with respondents revealed that older respondents were more likely to seek help and support with debugging from ChatGPT. We concluded that older workers attach more importance to receiving help and assistance on the job. The results also indicate that the effect of facilitating conditions is expected to increase with users' experience. Thus, when moderated by age and experience, facilitating conditions will have a significant influence on the adoption of ChatGPT for older workers, particularly with increasing experience.

4.5 Data Security

Data security is the practice of safeguarding digital data from unauthorized access, use, alteration, or destruction. It is crucial in protecting personal, confidential, or sensitive information from cyber threats such as data breaches, malware, and cyber-attacks. Our findings suggest that data security is a crucial factor in the adoption of ChatGPT for bug fixing.

When it comes to adopting ChatGPT for bug fixing, data security is a major concern for many professional developers. The data security service, Cyberhaven, detected and blocked requests from 4.2 percent of the 1.6 million workers at its client companies attempting to input data into ChatGPT due to concerns about the potential leakage of confidential information, client data, source code, or regulated information to unauthorized parties.

In our study, we found that 18 participants cited concerns about data confidentiality as the reason why they would not use ChatGPT to fix bugs. They were worried that the system might leak confidential information, putting their companies and customers at risk. Additionally, 8 out of 36 software engineers mentioned that they would not adopt ChatGPT because they do not want their inputs and outputs to be stored by the system and potentially distributed to other users later. These participants emphasized the importance of privacy and control over their data and were not willing to compromise on these principles.

However, 4 participants mentioned that they would still use ChatGPT for bug fixing, but would avoid copying and pasting the code directly due to concerns about potential security vulnerabilities. These participants preferred to manually review and modify the generated code to ensure that it was secure and reliable. 2 out of the 36 participants mentioned that functionality was the top priority in their companies, especially given the tight deadlines for delivering products. They were willing to adopt ChatGPT despite concerns about data security because they believed that it could improve productivity and speed up the bug fixing process.

4.6 Trust

We define trust as the extent to which an individual has confidence that ChatGPT can generate accurate and dependable responses.

Correctness is a significant factor for professional developers in deciding whether to adopt ChatGPT for bug fixing. Although ChatGPT can provide correct answers to many questions, the overall accuracy rate is still relatively low. Stack Overflow recently implemented a ban on ChatGPT due to its low success rate in generating correct answers. This ban was put in place because the posting of answers created by ChatGPT was deemed harmful to the site and its users who rely on correct answers to solve coding issues.

According to our study, 14 participants reported receiving incorrect responses from ChatGPT while fixing bugs. Out of these participants, 8 indicated that they would not use ChatGPT for bug fixing due to their lack of trust in its accuracy. 4 participants stated that they may continue to use ChatGPT in conjunction with other search engines to validate the information and ensure its reliability. While they expressed some concerns about ChatGPT's accuracy, they also acknowledged that ChatGPT does a good job in collecting information.

5. Conclusion

The emerging tool of ChatGPT has the potential to help professional software developers in fixing bugs, but it is not yet widely adopted. Through interviews with 50 professional developers and stakeholders, this study explores their attitudes and perceptions towards ChatGPT adoption for bug fixing. Building on the UTAUT theory, we identify several key factors that affect the adoption of ChatGPT for bug fixing, including performance expectancy, effort expectancy, social influence, facilitating conditions, data security, and trust. Our findings offer important insights into the factors that influence ChatGPT's adoption for bug fixing and can be used to guide future research in exploring additional potential factors. Overall, this study contributes to the understanding of the challenges and opportunities for using ChatGPT in software development, and could inform the development of strategies to promote its adoption.

Limitations and Future work

Our study has identified several key factors that can impact the adoption of ChatGPT for bug fixing by professional developers. These insights can provide valuable recommendations for ChatGPT developers and researchers to improve the technology. By considering the factors we have explored, ChatGPT can be improved to better meet the needs and preferences of professional developers, which can ultimately lead to increased adoption and usage. Our hope is that these recommendations will contribute to the continued development of ChatGPT and help ensure its successful adoption by professional developers.

However, it is important to acknowledge the limitations of our work. One major limitation is the relatively small sample size of participants, which may limit the generalizability of our findings. Our current project is a preliminary study that offers a first glimpse into the factors that impact the adoption of ChatGPT. In future work, we plan to use quantitative methods, including surveys, to draw more generalized conclusions that can be applied to a larger population of developers. This will provide a more comprehensive understanding of the factors influencing the adoption of ChatGPT for bug fixing.

Our interview study may be subject to threats to internal validity, which refers to the extent to which the study results can be attributed to the variables being studied rather than extraneous factors. One potential threat is social desirability bias, where participants may not provide truthful responses in order to present themselves in a favorable light. Despite our efforts to conceal our goal of potentially promoting the adoption of ChatGPT during the interview, there is still a possibility that participants could provide positive responses based on their assumptions about our intentions. Incorporating a survey study in our future research could be a useful approach to address the potential threats to internal validity. By using multiple methods to collect data, we can triangulate the findings and ensure consistency across different sources. Additionally, survey studies can help us reach a larger and more diverse sample of participants, reducing the potential for bias and increasing the generalizability of the results.

When considering using quantitative methods, it is essential to be aware of potential threats to validity, which can lead to inaccurate results. Construct validity is a potential threat to the validity of survey studies exploring the adoption of ChatGPT for bug fixing. This refers to the extent to which the survey questions and the responses provided by participants accurately measure the constructs of interest. In our interview study, we were able to clarify any questions that participants may have had, which helped to ensure that our study did not suffer from construct validity issues. However, when using surveys, if questions are not carefully designed or if participants misunderstand them, the resulting data may not accurately reflect their true attitudes or behaviors. It is therefore essential to carefully design survey questions to ensure they are clear and unambiguous and that participants understand them correctly. External validity is also a crucial factor to consider in survey studies examining the adoption of ChatGPT for bug fixing. External validity refers to the extent to which the study findings can be generalized to other populations and settings beyond the sample that was studied. Unlike our interview study where we had a small and diverse sample that could be easily controlled, survey studies can be affected by threats to external validity if the sample size is not representative of the broader population of software developers. This may lead to limited generalizability of the study results. Therefore, researchers should aim to recruit a representative sample of participants with diverse backgrounds to ensure generalizability of the findings. Survey studies can also face threats to internal validity, but because surveys are anonymous, the extent of the threats to internal validity by using surveys should be lower than in our interview study. Participants are more likely to answer questions honestly if they do not have to reveal their identity.

Another limitation of our study is that our study did not take into account the potential ethical implications of using AI-powered tools like ChatGPT in software engineering practice. While ChatGPT offers several benefits, such as improving efficiency, it may also result in job loss or a decrease in demand for software developers. This could potentially create a sense of resistance towards the adoption of ChatGPT by software developers. As such, future studies exploring the factors that influence the adoption of ChatGPT should take into account these ethical considerations and strive for a balanced approach that considers both the benefits and the potential drawbacks of using ChatGPT in software development.

Finally, it should be noted that our study focused solely on the scenario in which users input error messages and existing code, and then ask ChatGPT for suggestions on how to fix the errors. This was due to the nature of the interaction mode between ChatGPT and users. In the future, it would be worthwhile to investigate whether ChatGPT can improve its bug fixing ability by adding automation and breakpoints in the conversation. By exploring more scenarios, we can uncover additional possibilities for adopting ChatGPT for bug fixing in software engineering practices.

References

[1] Xiangyu Li, Shaowei Zhu, Marcelo d'Amorim, and Alessandro Orso. 2018. Enlightened debugging. In Proceedings of the 40th International Conference on Software Engineering. 82–92.

[2] Min Xie and Bo Yang. 2003. A study of the effect of imperfect debugging on software development cost. IEEE Transactions on Software Engineering 29, 5 (2003), 471–473.

[3] Strategic Planning. 2002. The economic impacts of inadequate infrastructure for software testing. National Institute of Standards and Technology 1 (2002).

[4] Abid Haleem, Mohd Javaid, and Ravi Pratap Singh. 2023. An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges. Bench Council Transactions on Benchmarks, Standards and Evaluations (2023), 100089.

[5] Nigar M Shafiq Surameery and Mohammed Y Shakor. 2023. Use ChatGPT to solve programming bugs. International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290 3, 01 (2023), 17–22.

[6] Dominik Sobania, Martin Briesch, Carol Hanna, and Justyna Petke. 2023. An Analysis of the Automatic Bug Fixing Performance of ChatGPT. arXiv preprint arXiv:2301.08653 (2023).

[7] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. 2003. User acceptance of information technology: Toward a unified view. MIS quarterly (2003), 425–478.

[8] Tore Dybå and Torgeir Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. Information and software technology 50, 9-10 (2008), 833–859.

[9] Pavneet Singh Kochhar, Tegawendé F Bissyandé, David Lo, and Lingxiao Jiang. 2013. An empirical study of adoption of software testing in open source projects. In 2013 13th International Conference on Quality Software. IEEE, 103–112.

[10] Hart O Awa, John P Uko, and Ojiabo Ukoha. 2017. An empirical study of some critical adoption factors of ERP software. International Journal of Human– Computer Interaction 33, 8 (2017), 609–622.

[11] Everett M Rogers, Arvind Singhal, and Margaret M Quinlan. 2014. Diffusion of innovations. In An integrated approach to communication theory and research. Routledge, 432–448.

[12] Shundan Xiao, Jim Witschey, and Emerson Murphy-Hill. 2014. Social influences on secure development tool adoption: why security tools spread. In Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing. 1095–1106.

[13] Hemank Lamba, Asher Trockman, Daniel Armanios, Christian Kästner, Heather Miller, and Bogdan Vasilescu. 2020. Heard it through the Gitvine: an empirical study of tool diffusion across the npm ecosystem. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 505–517.

[14] Linda G Wallace and Steven D Sheetz. 2014. The adoption of software measures: A technology acceptance model (TAM) perspective. Information & Management 51, 2 (2014), 249–259.

[15] Martin Fishbein. 1979. A theory of reasoned action: some applications and implications. (1979).

[16] Richard M Ryan and Edward L Deci. 2000. Intrinsic and extrinsic motivations: Classic definitions and new directions. Contemporary educational psychology 25, 1 (2000), 54–67.

[17] Icek Ajzen. 1991. The theory of planned behavior. Organizational behavior and human decision processes 50, 2 (1991), 179–211.

[18] Shirley Taylor and Peter A Todd. 1995. Understanding information technology usage: A test of competing models. Information systems research 6, 2 (1995), 144–176.

[19] Ronald L Thompson, Christopher A Higgins, and Jane M Howell. 1991. Personal computing: Toward a conceptual model of utilization. MIS quarterly (1991), 125–143.

[20] A. Bandura. 1986. Social foundations of thought and action: A social cognitive theory. Prentice Hall, Englewood Cliffs, NJ.

[21] C.B. Seaman. 1999. Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering 25, 4 (1999), 557–572. https://doi.org/10.1109/32.799955

[22] Jonathan Smith, Paul Flowers, and Michael Larkin. 2009. Interpretative Phenomenological Analysis: Theory, Method and Research. Qualitative Research in Psychology 6 (01 2009).

[23] Bambang Leo Handoko. 2019. Application of UTAUT theory in higher education online learning. In Proceedings of the 2019 10th International Conference on E- business, Management and Economics. 259–264.

A Interview questions

Below are the interview questions that we asked during the interviews for your reference:

A.1 General questions

1) Do you know about ChatGPT?
2) Have you used it before to fix bugs?
3) Will you consider using it to fix bugs in the future?
4) Why or why not you will adopt it in the future?

A.2 Performance Expectancy

1) In your experience, do you believe that using ChatGPT for bug fixing can improve job performance? Why or why not?
2) In your opinion, does using ChatGPT for bug fixing enhance productivity? Why or why not?
3) Do you find that ChatGPT helps you accomplish tasks more quickly? Why or why not?
4) Would these factors influence your decision to adopt ChatGPT? Why or why not?

A.3 Effort Expectancy

1) In your experience, do you find ChatGPT to be user-friendly? Why or why not?
2) Do you find it easy to learn how to use ChatGPT? Why or why not?
3) Would these factors influence your decision to adopt ChatGPT? Why or why not?

A.4 Social Influence

1) Would you be inclined to use ChatGPT for bug fixing if it were recommended to you by your colleagues or team leader? Why or why not?

A.5 Facilitating Conditions

1) Do you feel that you have the necessary resources to effectively use ChatGPT?
2) Do you feel that you have the necessary knowledge to effectively use ChatGPT?
3) Would these factors influence your decision to adopt ChatGPT? Why or why not?