# Enhancing Debugging Efficiency in Software Development through AI-powered Intelligent User Interfaces (IUIs)

A Group Study

Candidate no: 575

# Contents

# List of Figures

# 1. Introduction

In the world of software development there are a many tools that are used to speed up the development processes. However one part of this process that has stayed resistant to change is debugging. Debugging in the sense of software development is correcting errors whether they are of a syntax nature or logical type. Programming languages adhere to strict rules ranging from allowed whitespace to closing brackets correctly. Navigating through the vast minefield of errors a simple oversight can make is a tedious and never ending task. This is coupled with some times vague error messages that don't always tell you what is wrong. This is especially true for errors of the logical type. Where everything is written correctly but there is a logical error that is occurring somewhere making the program behave in a certain way. To make this even more cryptic and hard to decipher there are errors that only show up during certain conditions. Making them harder to weed out. This is where Artificial Intelligence (AI) can be of immense help.

First regarding syntax errors, the nature of LLMs make them able to quickly go through the code and update all the syntax mistakes made as seen by commercially available tools such as Github Copilot (Github, n.d.). The AI would understand the error code and make the needed adjustments. Making such errors a thing of the past.

Secondly if we look at logical errors, the AI would be able to better understand and explain the error message and give suggestions to how to resolve this. Making the process much smoother and simpler for the developer. Regarding those hard to come by errors that only show up during certain conditions, it could be more efficiently traced with an AI that can see any logical fallacies that the developer might miss.

Debugging is a resource intensive and tedious process that could be better served by adapting some the AI tools available in the marked today.

With the advent of GPT 3.5 LLM's have taken the world with storm. They offer a very fast and flexible implantation to already existing developer tools. The likes of Github's Copilot and Replit have made it much simpler to use AI tools in a user-friendly way that fosters fluid collaboration between human and AI.

In this study we wish to see how this interaction takes place. By inviting participants to solve debugging challenges and provide a AI-powered code editor that they can interact with we can try to answer the question of how can AI-powered Intelligent User Interfaces (IUIs) enhance the efficiency and effectiveness of debugging processes in software development.

# 2. Background

## 2.1 Existing Debugging Tools and Techniques

Debugging is a consequence of humans being the ones making software. We make mistakes and therefore debugging is needed. In order to help with this process several different tools exist. We have the likes of IDE's that include debuggers. Debuggers allow the user to breakdown the code and inspect line by line how it is executing. One can inspect what variables are saved and memory usage. All of these things help with reviewing ones code (Beller et al., 2018).

There is also a realm of different logging tools, that monitor the software running so that when and if something goes wrong one can get a better idea of what might have caused or what conditions led to a systems failure (Jiang et al., 2023).

Overall these tools are meant to help implement certain techniques that are well used in order to solve the issues at hand. The techniques mentioned in (Cornell University, n.d.) are some of these. They can be loosely grouped into errors message analysis, automated testing and root cause analysis. All these techniques are there in order to facilitate a rigorous way of weeding out errors.

All of these tools are here to facilitate error handling and debugging software. However they still require a manual execution of correcting said error. Some of the limitations of these tools are down to the complexity of using them as stated in the (Beller et al., 2018), lack of education makes using debuggers specifically in IDEs hard but this can be extrapolated to other tools as well. The manual nature of debugging might be the issue here and the overall limiting factor. As also mentioned in the study a lot of the testing that is done doesn't necessarily reduce debugging, given that the test code might need debugging as well, creating a everlasting loop.

## 2.2 Emergence of AI

As the rise of LLMs have emerged different ways to incorporate this into our lives are always being proposed. One of the more newer ones and most prominent to take the power of these LLMs where Github and their own unique adaption of the GPT-3.5 model made by Open AI (OpenAI, n.d.). This version of GPT-3.5 is names codex and is to work inside ones code editor. It does things like code completion, code suggestions and even being able to write complete programs to some extent (Github, n.d.). LLMs are AI trained on a large set of text that enables it to produce new text that feels almost humanlike (Shanahan, 2023). Taking the capabilities of these LLMs they have created codex which is trained on the huge repositories of code that is public inside of Github (OpenAI, n.d.). Allowing the model to be able to produce code and use the context of the code to produce and control the code being worked on.

The major benefits one gets is now a software developer can interact with an

AI model that has the ability to both create and verify code in mere seconds. The way LLMs work is that they don't understand what they are outputting, but they are able to generate code that is statistically most like to correspond to what you are working on, meaning that all the training its has had will likely give you a correct output. But in here also lies its limiting factor. Given that it doesn't understand, it will only be able to mimic human intelligence (Shanahan, 2023).

# 3. Methodology

## 3.1 Participant Selection and Demographics

When choosing participants we decided that the nature of the challenges presented need some level of familiarity with programing. Therefore all the participants had some previous experience with programming and debugging. In order to simplify the recruitment process we choose to go with people we knew. This where friends and family that we could reach out to that would not require any major incentive to participate.

Most of the group members are male, and proportionally most of the participants are also male. The age range also falls thereafter. We did not decide on excluding anyone and the only requirement was that they had some experience with programming.

In the pre-survey we decided to get generic user data, like gender, age, role and familiarity with python and AI tools. This data was mainly gathered to get a better sense of the participants and to better understand the context for their answers.

## 3.2 Data Collection Tools: Surveys and Interviews

For this study we conducted a pre and post-survey, with a interview following the surveys. This was seen as the best way to get both qualitative data and quantitive data, where we could also engage with the participants and probe deeper.

We used a google form to collect all the data. This was chosen as it was what we were familiar with from before. The surveys and interview was contained in a single form, meaning that the participants answered all the questions in one sitting. This was done in order to avoid GDPR constraints of tracking participants. No recordings either voice or video was made of the participants.

## 3.3 Study Design: Debugging Tasks and Tool Usage

Assessing how a participant approaches debugging a piece of code can be challenging in the sense that we want to know how debugging tools are used rather then the debugging its self. So we decided that we would make the problems less complex in nature and have two codes for review. Code one is a runtime error, that gives a 'ZeroDivisionError'. The second piece of code will raise an 'IndexError'. The challenges are divided up into easy and hard. This is subjective, but we decided that we wanted a simple one that could be done with no tools and one that might require a debugging tool. Below are the code challenges given to the participants:

```python
# code challenge 1:
a = 45
b = 0
result = a / b
print(result)

# code challenge 2:
class Solution:
    def isPalindrome(self, x: int) -> bool:
        stringValueOfX = str(x)
        reverseValueOfX = ""

        for i in range(len(stringValueOfX), -1, -1):
            reverseValueOfX += stringValueOfX[i]

        return reverseValueOfX == stringValueOfX
```

The challenges were setup using Replit. Replit is an online cloud code editor that allows users to code, run and deploy software created. It includes different tools, including an AI-powered chat and an inline AI code completion and suggestion component (Replit, n.d.). While Replit can be used as a free service in order to access all features including the AI module, a monthly subscription is required. When signing up when gets a 14 day free trial which is why we decided to use this service. Replit uses the same LLM as GitHub's copilot and is therefor based on the aforementioned codex. This would be considered among the most prominent and well known LLMs commercially available to end users.

The participant were shown a video tutorial of how Replit is used and how they can access the AI module. The main focus here was the AI-powered chat function which allows the percipients to either copy and paste the code into the chat or just ask what the error message means and how to resolve. They are then given the instructions that they can use other tools to solve the challenges. This was done in-order to see how incline they were to use AI-powered debugging in their workflow. They were then given no time limit and they were to work on the problems. After they had successfully completed one challenge we inspected the code and checked if it was correct. We then moved on to challenge number two and repeated the process. When both challenges where completed we conducted the post-survey and the interview.


Figure 1: Replit code editor with AI chat activated

# 4. Survey and Interview Analysis

Before the challenges where issued we conducted a pre-experiment-survey. The questions in this survey were there to get general information about the percipients and also to see how they felt about AI in general and in the context of debugging. The complete survey can be seen in appendix A.

The pre-survey revealed that most of the participants had some familiarity with both AI in general and traditional debugging methods. What was interesting to see was that most of the participants said that traditional debugging methods where quite efficient, yet believed that AI-powered debugging through a chat interface would be helpful. This is even more reinforced by the fact that the participants believe that AI-powered chat interface will facilitate better collaboration during the debugging process.

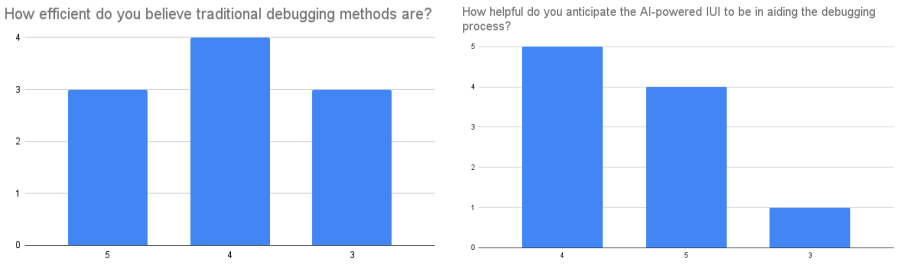


Figure 2: How efficient do you believe traditional debugging methods are?/How helpful do you anticipate the AI-powered IUI to be in aiding the debugging process?

After the conclusion of the challenges we conducted a post-survey interview of the participants. Here we were looking to see how they felt about using Replit and its AI-powered chat, and to see if they managed to complete the tasks. Most of the participants felt it was a useful aid in the challenges. Some however found it had a step learning curve. This might be a reflection of our tutorial not being good enough, however most of the participants found it easy to use. Almost all the participants managed to complete both challenges.

Further analyzing the pre and post-survey we conduct a paired t-test calculation. A paired t-test calculation is used to compare two data points. It gives a statistical view of how they are correlated and if there is significant differences between them (Yeager, n.d.). In this calculation we looked at the anticipated and the percived helpfulness of the AI-powered chat interface in regards to debugging. Given that the p-value is above the standard threshold (0.05) for the null-hypothesis this indicates that the anticipated helpfulness of using AI-powered chat interface for debugging was meet, $t(9) = 0.43, p = 0.68$. Here the null-hypothesis would be the anticipated helpfulness of the AI-powered chat interface.

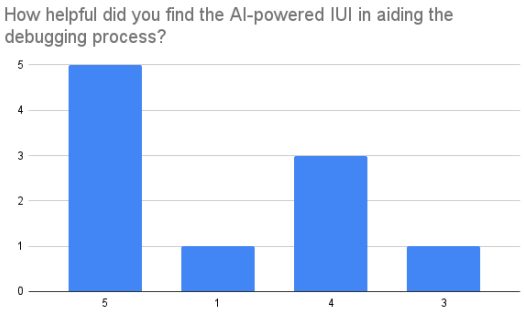


Figure 3: How helpful did you find the AI-powered IUI in aiding the debugging process?

Analyzing the interview one can see that there are several themes that emerge. We can categorize them in to the following:

1. Efficiency and speed
2. Intuitiveness and usability
3. Accuracy and reliability
4. Future willingness to use

We can see that the questions asked facilitated these topics to be brought up and we can see summarize from the answers that the participants overall where positive booth to the experience of using the AI-powered chat interface to debug the code, and that they believe that it would be beneficial to add to their existing workflow.

If we dive deeper into the different themes we can try to pick up some of the underlying sentiments.

***Efficiency and speed***

The participates felt mostly that it increased their speed and increased their efficiency. They emphasized that the AI-powered chat helped and was superior to their traditional methods of debugging. This indicates that the value of time saving is appreciated. There were cases of frustration when the AI-powered chat gave unclear answers or wasn't faster then the manual debugging.

***Intuitiveness and usability***

The answers related to this came in a bit varied. Some enjoyed the interface and felt it was easy and intuitive to use while others had a harder time. For those that felt it was easy to use, several had previously used similar tools and where familiar with the interface style, and they indicated a strong positive sentiment. While the ones that indicated a more negative feeling also hadn't used it before. It's worth noting that some of the participants chose to continue debugging the challenges on other platforms. These gave the indication that they felt more

comfortable in their preferred setup. One included a combination of vs code and chat-gpt and an other used stackoverflow to get a solution to the error message.

***Accuracy and reliability***

Most of the participants said that they received the correct answer when they interacted with the AI, this might have given a positive sentiment towards the usage of the chat interface. There were concerns regarding trust of the results. Even after the AI gave the correct answer some of the participants where hesitant to deem it completely trustworthy. One of the participants stated that they didn't not know where the information was coming from. Indicating a low level of trust.

***Future willingness to use***

Despite the concerns raised by some of the participants, all unanimously said that they where positive to integrating an AI-powered chat interface into their workflow. Some cited the increase in efficiency and others highlighted its usability. Overall the sentiment to the use of such tools where a welcomed improvement over the traditional methods of debugging.

The interview analysis showed that most of the participant where in alignment with their answers. There was one answer that seemed out of place. Responding to the question "Would you be willing to use the AI-powered Interface for future debugging tasks? Why or why not?", they responded with "Yes, because I trust it and it provides fast solutions and reduces the amount of time spent.". This level of trust displayed was surprising, yet the value in time saving seems to shine through.

# 5. Discussion

## 5.1 Interpretation of Findings

Looking at the overall analysis of the participants responses the positive sentiment towards using AI-powered chat interface to debug code seems to be apparent. This positive response lens to the notion that combining AI with Human-Computer interaction (HCI) in this context could be beneficial. The presence of a chat interface encourages a dialog like communication with the AI. This is a familiar type of communication for most people given that it mimics human to human correspondence. This is further reinforced by other studies conducted in this field (Ross et al., 2023).

As with all new technology a level of uncertainty is inherently baked into the use. This is no different regarding AI powered tools. What we saw from the responses is that the usability and reliability plays a factor in the level of trust that can be ascribed. The more usefull and correct responses one receives the more likely one is to trust it. As can be seen by this study from KPMG done in 2023 (Gillespie et al., 2023), here they show that trust level is related to perceived benefits. This is also what we saw from our participants. Even though trust levels varied, they all stated they where positive to implementing such tools into their debugging workflow. The reasons given reflected the perceived value of time saved.

The efficiency and speed that AI-powered chat interface gave is the main reason we see for adoption of such tools. Even if there is a lack of trust. As long as it saves time it will be valuable to some extent. This is a sentiment that is broadly excepted as the advent of AI becomes more prevalent. In the aforementioned study from KPMG this sentiment was stressed, the benefits such as increased speed and efficiency will for some be worth the risk. However there seems to be a first impression bias. Some of the participants as noted switched tools in order to use tools that they were more comfortable using. This indicates that if they don't receive the perceived value of time saving fairly quickly they might disengage with the AI tool. Broadly speaking this might be extrapolated to AI-powered tools in general. If it doesn't bring value instantly the lack of trust inherent in new technology might win out.

As we moved forward towards an AI centric world. More and more tools will be developed to improve the difference steps in the software development life cycle. Debugging being one of the more critical components in this life cycle we believe more attention will be given to how we can better use AI-powered tools to better this process. In order for AI-powered tools in general to be efficient, usability needs to be in-focus. As we saw with some of the participants, they felt that the learning curve was a bit to steep. This results in more friction then necessary. This revelation highlights that HCI is as important for adoption of AI-powered tools as the AI component.

## 5.2 Limitations and Future Work

AI-powered chat interface used in the context of debugging opens up a wide range of opportunities. However as mentioned before the level of trust one is to give these tools seems to be a sticking point. Raised in the interview was the fact that not knowing where the information that the AI provides is cause for a pause. This seems to be the one of the major limiting factors of such tools. An other limiting factor is the usability. It was said by a participant that without the tutorial video, they wont not have been able to locate the AI module inside of Replits code editor. Removing this barriers of interacting with the AI, will need to be a priority in the future.

To further see the benefits of using such AI-powered chat interface for debugging purposes a larger study should be conducted with a more varied participants. This would capture a broader perspective and could be insightful in-order to asses what the most important factors such a tool would need in-order to appeal to a wide range of users.

# 6. Conclusion

In this study we conducted a survey pre and post, debugging challenges that participants where asked to complete. They used Replits AI-powered chat interface in-order to correct the mistakes in the code. After the surveys we sat down with the participants and conducted an interview to learn more about how they felt about the Intelligent User Interface (IUI) that they hade interacted with.

The overall response where positive. Most of the participants used the AI tool to complete the challenges, however some reverted back to their usual tools where they felt more comfortable. Lack of trust and efficiency was sited as main reasons.

Most of the participants felt that they worked better with the AI tool, even though complaints about the usability was raised, a steep learning curve, was cited as an issue.

The overall sentiment and conclusion revealed that all participants would like to introduce a similar tool into their workflow, given the perceived benefits it gave them with regards to speed and efficiency.

Concluding this report we see that there is a place for AI-powered tools to improve upon the traditional debugging tools that exist. However there needs to be an emphasis on usability and increase in trust level, by being more reliable. In order to achieve great tools with AI, both the I and UI in Intelligent User Interface (IUI) has to be equally balanced.

# 7. References


Beller, M., Spruit, N., Spinellis, D., & Zaidman, A. (2018). On the dichotomy of debugging behavior among programmers. *Proceedings of the 40th International Conference on Software Engineering*, 572–583. https://doi.org/10.1145/3180155.3180175

Cornell University. (n.d.). *Lecture 26: Debugging Techniques*. https://www.cs.cornell.edu/courses/cs312/2006fa/lectures/lec26.html.

Gillespie, N., Lockey, S., Curtis, C., Pool, J., & Ali Akbari. (2023). *Trust in Artificial Intelligence: A global study*. The University of Queensland; KPMG Australia. https://doi.org/10.14264/00d3c94

Github. (n.d.). GitHub Copilot · Your AI pair programmer. In *GitHub*. https://github.com/features/copilot.

Jiang, P., Sun, F., & Xia, H. (2023). Log-it: Supporting Programming with Interactive, Contextual, Structured, and Visual Logs. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–16. https://doi.org/10.1145/3544548.3581403

OpenAI. (n.d.). *OpenAI Codex*. https://openai.com/blog/openai-codex.

Replit. (n.d.). *Build, ship, and share software with Replit's powerful IDE - Replit*. https://replit.com/site/ide.

Ross, S. I., Martinez, F., Houde, S., Muller, M., & Weisz, J. D. (2023). The Programmer's Assistant: Conversational Interaction with a Large Language Model for Software Development. *Proceedings of the 28th International Conference on Intelligent User Interfaces*, 491–514. https://doi.org/10.1145/3581641.3584037

Shanahan, M. (2023). *Talking About Large Language Models* (arXiv:2212.03551). arXiv. https://arxiv.org/abs/2212.03551

Yeager, K. (n.d.). *LibGuides: SPSS Tutorials: Paired Samples t Test*. https://libguides.library.kent.edu/SPSS/PairedSamplestTest.

# 8. Appendices

## Appendix A: Survey Questionnaires

***Pre-survey***

| Participant ID | Age | Familiarity with Python |
|---|---|---|
| 1 | 29 | 3 |
| 2 | 28 | 3 |
| 5 | 30 | 5 |
| 6 | 27 | 3 |
| 11 | 37 | 4 |
| 12 | 21 | 5 |
| 21 | 25 | 3 |
| 22 | 28 | 5 |
| 41 | 25 | 3 |
| 42 | 26 | 3 |

| Participant ID | Familiarity with AI Tools |
|---|---|
| 1 | 4 |
| 2 | 2 |
| 5 | 3 |
| 6 | 3 |
| 11 | 5 |
| 12 | 3 |
| 21 | 2 |
| 22 | 3 |
| 41 | 2 |
| 42 | 4 |

| Participant ID | How comfortable do you feel with traditional debugging methods? |
|---|---|
| 1 | Very Comfortable |
| 2 | Somewhat Comfortable |
| 5 | Very Comfortable |
| 6 | Neutral |
| 11 | Somewhat Comfortable |
| 12 | Somewhat Uncomfortable |
| 21 | Somewhat Comfortable |
| 22 | Very Comfortable |
| 41 | Very Comfortable |
| 42 | Neutral |

| Participant ID | How efficient do you believe traditional debugging methods are? |
|---|---|
| 1 | 5 |
| 2 | 4 |
| 5 | 5 |
| 6 | 4 |
| 11 | 3 |
| 12 | 3 |
| 21 | 4 |
| 22 | 5 |
| 41 | 3 |
| 42 | 4 |

| Participant ID | How helpful do you anticipate the AI-powered IUI to be in aiding the debugging process? |
|---|---|
| 1 | 4 |
| 2 | 4 |
| 5 | 5 |
| 6 | 5 |
| 11 | 4 |
| 12 | 4 |
| 21 | 5 |
| 22 | 4 |
| 41 | 3 |
| 42 | 5 |

| Participant ID | How easy do you anticipate it will be to learn to use the AI-powered IUI? |
|---|---|
| 1 | 5 |
| 2 | 4 |
| 5 | 5 |
| 6 | 4 |
| 11 | 5 |
| 12 | 4 |
| 21 | 3 |
| 22 | 1 |
| 41 | 4 |
| 42 | 4 |

| Participant ID | Do you believe the AI-powered IUI will facilitate collaboration during the debugging process? |
|---|---|
| 1 | 5 |
| 2 | 4 |
| 5 | 5 |
| 6 | 5 |
| 11 | 5 |
| 12 | 4 |
| 21 | 5 |
| 22 | 4 |

***Post-Survey***

| Participant ID | How helpful did you find the AI-powered IUI in aiding the debugging process? |
|---|---|
| 1 | 5 |
| 2 | 5 |
| 3 | 5 |
| 4 | 1 |
| 5 | 4 |
| 6 | 5 |
| 7 | 4 |
| 8 | 5 |
| 9 | 4 |
| 10 | 3 |

| Participant ID | How easy was it to use the AI-powered IUI for debugging? |
|---|---|
| 1 | 1 |
| 2 | 4 |
| 3 | 5 |
| 4 | 3 |
| 5 | 4 |
| 6 | 4 |
| 7 | 5 |
| 8 | 5 |
| 9 | 1 |
| 10 | 3 |

| Participant ID | How steep was the learning curve to effectively use the AI-powered IUI for debugging? |
|---|---|
| 1 | 3 |
| 2 | 5 |
| 3 | 5 |
| 4 | 4 |
| 5 | 3 |
| 6 | 2 |
| 7 | 4 |
| 8 | 5 |
| 9 | 1 |
| 10 | 2 |

| Participant ID | Did the AI-powered IUI facilitate collaboration during the debugging process? |
|---|---|
| 1 | Yes |
| 2 | Yes |
| 3 | Yes |
| 4 | No |
| 5 | Yes |
| 6 | No |
| 7 | Yes |
| 8 | Yes |
| 9 | Yes |
| 10 | Yes |

| Participant ID | How many challenges did you complete? |
|---|---|
| 1 | 2 |
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 2 |
| 9 | 2 |
| 10 | 2 |

## Appendix B: Interview Transcripts

| **What tools did you use to solve the challenges?** | Participant ID |
|---|---|
| AI tool provided by the Replit.com | 1 |
| VS code | 5 |
| ChatGPT and Visual Studio Code | 6 |
| Stackoverflow | 21 |
| AI | 41 |
| I just used the AI provided | 42 |
| AI | 2 |
| I used AI IUI to solve the two challenges | 11 |
| For the 1st Challenge I used line by line debuggingFor the 2nd Challenge I used Replit Ai in collaboration with my traditional debugging skills | 12 |
| Manual debugging | 22 |

| **Tell me more** | Participant ID |
|---|---|
| Debugger tool provided by the Replit.com | 1 |
| Visual Studio Code is my preferred programming tool. And ChatGPT is the AI I have the most familiarity with. | 6 |

| **Why did you choose those tools?** | Participant ID |
|---|---|
| I wanted to debug the programs using AI tool in order the understand the efficiency and usefulness of the tool. | 1 |
| I am comfortable using VS code because I have past experience and user friendly. | 5 |
| Familiarity with both of them and trust | 6 |
| Stackoverflow is the first ranked page that came up after searching the error | 21 |
| It seemed the simplest. Given that it was right there | 41 |
| I tried it on the first task and gave me immediate correct feedback, so i keept using it | 42 |
| It gives detailed feedback on your whole code | 2 |
| I choose the AI tools to experiment how effective it is to solve technical challenges | 11 |
| Python run line by line so the easy method to debug it is to review code line by line | 12 |
| That is the tool I am comfortable with the most | 22 |

| **Tell me more** | Participant ID |
|---|---|
| To compare the AI tool with the traditional debugger, I used Debugger tool as well. | 1 |
| Visual Studio Code is user-friendly. And so is ChatGPT. I have familiarity with both of them. I trust ChatGPT and it saves time. | 6 |

| **Did you find the AI-powered Interface helpful in identifying bugs/errors? Please explain.** | Participant ID |
|---|---|
| The programs were easy to understand, and debugging the codes was straightforward using both AI-powered Interface and traditional debugger. | 1 |
| Yes. It was very helpful in providing the solution and explaining errors in the code. | 5 |
| Yes I did. ChatGPT gives a theoretical answer as well as an example for the solution of the problem. | 6 |
| No, there was no suggestion from Replit on which variable to change on the first and second solution | 21 |
| Yes, it gave me an understanding of the error and solution. | 41 |
| Yes, it gave me both the correction and explanation making it easy to see the mistake | 42 |
| Yes. In this challenge, I got divide by 0. | 2 |
| Yes, the interface is friendly and I found it helpful | 11 |
| Yes. It indicated where the error was coming from specifically the line and the function that has a bug in it | 12 |
| No, while I found it useful in giving more context and explanation to the bugs, it didn't help in identifying. | 22 |

| **Tell me more** | Participant ID |
|---|---|
| AI-powered Interface was really helpful to find the bugs, though, there was only one bug/improvement suggested by the AI-powered Interface. Using the tool was a great experience, as the instructions were detailed and to the point. | 1 |
| Yes. It was very helpful in providing the solution and explaining errors in the code. | 5 |
| No I did not, just like the Python console, after running the code on Replit, I found out exactly the same message as I was getting on the Python console. | 21 |
| Yes, by 10-15 min | 41 |
| No I did not. | 22 |

| **Did the AI-powered Interface speed up the debugging process for you? If so, can you estimate by how much?** | Participant ID |
|---|---|
| Definitely, it was quick to debug the programs using AI-powered Interface. Before, using the interface, I had to look into each line of the code to look for bugs and errors. But, the tool made the process easy, it showed where the bug is and what to improve. | 1 |
| Yes. I was able to resolve the error in less than 10 minutes. | 5 |
| Yes, 25 min for problem number 2 | 6 |
| No it did not, I was expecting the Replit console to make it easier for me since it is AI powered but I spent as much time on first question with both manual and the AI powered interface. | 21 |
| Yes, by 10-15 min | 41 |
| Yes, i would say it did. Maybe 2x faster | 42 |
| More than 50% | 2 |
| Yes, within 5 secs | 11 |
| Yes. Let's say approximately 4-5 mins faster | 12 |
| Both codes were debugged with manual debugging, but using the AI chat tool on Replit did not speed up the process as significantly as expected | 22 |

| **Tell me more** | Participant ID |
|---|---|
| 25 min for problem number 2 | 6 |

| **How intuitive did you find the interface? Were there any features that you found particularly easy or difficult to use?** | Participant ID |
|---|---|
| It was really intuitive, I tried the AI feature and Debugger, and both of them were easy to use. They were detailed with outcomes and solutions. | 1 |
| It was a bit difficult for me as a first user but I was able to use it after spending 10 to 15 minutes playing around with the interface and navigating through the options. | 5 |
| Very intuitive. | 6 |
| The interface was intuitive for autocompletion, but not for automatically checking and debugging an existing code. | 21 |
| Fairly simple and very easy to navigate | 41 |
| The interface was somewhat simple, it is similar to ChatGPT. | 42 |
| Very intuitive. Asking questions and having feedback. | 2 |
| Very intuitive, no features are difficult, all are easy to use | 11 |
| The interface is great. but I believe the TOOLS icon should be in a visible sidebar | 12 |

| **How intuitive did you find the interface? Were there any features that you found particularly easy or difficult to use?** | Participant ID |
|---|---|
| The AI chatbot tool was intuitive and easy to use. | 22 |

| **Tell me more** | Participant ID |
|---|---|
| ChatGPT allows for direct posting of the code into the message box, and it will give a reply with a possible solution. | 6 |

| **Did you feel there was a learning curve to effectively use the AI-powered Interface for debugging? How steep was it?** | Participant ID |
|---|---|
| It was a new experience for me, as I tried AI-powered Interface for the very first time. It was easy to learn and use effectively. | 1 |
| Yes, around 0.3. | 5 |
| No | 6 |
| Yes I think I need more time to understand how Replit works. | 21 |
| Ja, but very low | 41 |
| No, not really. There were a lot of things going on the screen, but the tutorial helped a lot. I wouldn't have found the AI button so easily without it. | 42 |
| Very easy | 2 |
| Yes | 11 |
| It was easy to use, the learning curve wasn't steep at all. | 12 |
| Not much learning curve since I am familiar with AI tools before now | 22 |

| **Tell me more** | Participant ID |
|---|---|
| I am familiar with the use of ChatGPT. | 6 |

| **How would you compare your experience with the AI-powered Interface to traditional debugging methods?** | Participant ID |
|---|---|
| I am really satisfied with this, it took less time to debug the program compared to the conventional debugger. | 1 |
| The AI made the task much easier than the traditional method. | 5 |
| Simpler and yet more dangerous. | 6 |
| I did not notice much difference. | 21 |

| **How would you compare your experience with the AI-powered Interface to traditional debugging methods?** | Participant ID |
|---|---|
| Simpler and yet more dangerous. | 41 |
| It made the process simpler and more focused. I just need to go to the chat and it will tell me exactly what i did wrong. | 42 |
| AI is more faster than traditional debugging | 11 |
| AI-powered debugging is much faster than traditional debugging methods because it points out where the bug is coming from whereas in traditional method you have to review your entire codebase. | 12 |
| Traditional debugging methods seem faster and come more naturally than the AI-Powered interface | 22 |

| **Tell me more** | Participant ID |
|---|---|
| When you always run to the AI you miss the ability to learn as a dev. The time spent on a problem is a huge part of the process of learning. This more so for students and newer devs than more senior, but there is always something to learn. | 6 |

| **Dangerous how?** | Participant ID |
|---|---|
| When you always run to the AI, you miss the ability to learn as a dev. The time spent on a problem is a huge part of the process of learning. This is more so for students and newer devs than more senior, but there is always something to learn. | 41 |

| **Did you notice if the AI-powered Interface accurately identified bugs/errors? Were there any false positives/negatives?** | Participant ID |
|---|---|
| There was no actual error in the programs, but there was an improvement, and AI-powered Interface was accurate to suggest the interface. There was no false positive/negative. | 1 |
| Yes. There was no false bugs. | 5 |
| It accurately identified the bug and error | 6 |
| Based on my test, I think the AI interface did not indicate where the bug is in the code as I had expected. | 21 |
| It accurately identified the bug and error | 41 |
| It gave the correct answer on the first try | 42 |
| Sometimes it depends on the data. | 2 |

| **Did you notice if the AI-powered Interface accurately identified bugs/errors? Were there any false positives/negatives?** | Participant ID |
|---|---|
| Yes, it accurately debugs it, no negatives | 11 |
| The results I received were mostly positive results. | 12 |
| Yes, the AI-powered tool accurately identified the errors | 22 |

| **Tell me more** | Participant ID |
|---|---|
| It accurately identified errors. There were no false positives. | 6 |

| **Would you be willing to use the AI-powered Interface for future debugging tasks? Why or why not?** | Participant ID |
|---|---|
| Definitely, I will use the AI-powered Interface for debugging tasks in future, as it is less time consuming, easy to use and efficient to find the bugs. | 1 |
| Yes, because I trust it and it provides fast solutions and reduces the amount of time spent. | 5 |
| Yes, I use it in my daily work already. It's faster than googling. | 6 |
| Yes, I would be willing to use the AI-powered interface, however not for tasks that require immediate completion. | 21 |
| Yes, however, trust might be an issue. | 41 |
| Yes, because it eases the debugging task so easily. | 42 |
| Yes, because it increases my speed and helps me debug my codes faster. | 2 |
| Yes, for bugs that are difficult to manually debug. | 11 |
| Yes, for bugs | |

| **How so?** | Participant ID |
|---|---|
| Because I don't know where the AI gets its information from, but then again, I don't know where people on Google get their information. I think blindly following anything is a problem, and we as students do that a lot. | 42 |

| **How do you use?** | Participant ID |
|---|---|
| I ask it questions and use it as a starting point. Helps me figure out what I'm supposed to do and make a plan. | 41 |

| **Tell me more** | Participant ID |
|---|---|
| It saves a lot of time. It teaches well if you have coding questions. And the solutions are often very good. | 6 |