



Incorporating ChatGPT in Software Project Management*

Qamar Abbas, Waqas Younus, Shumain Malik, Muhammad Hammad Hassan

¹UBIT, University of Karachi, Sindh, Pakistan; ²Law School, Zhongnan University of Economics and Law, Wuhan, China; ³Law School, Zhongnan University of Economics and Law, Wuhan, China; ⁴UBIT, University of Karachi, Sindh, Pakistan. ;
Email: qmarabbas715@gmail.com, muhammadhammadhassan002@gmail.com, waqasyounus643@gmail.com, shumainMalik@gmail.com,

ABSTRACT

This paper investigates the applications of Chat-GPT, a state-of-the-art language model, in the field of applied software project management. Chat-GPT, developed by OpenAI, is a generative AI model that has demonstrated remarkable advancements in natural language processing and generation of text based output. This paper aims to highlight the potential uses of Chat-GPT in project management, including communication, risk management, resource allocation, and decision-making. By leveraging the capabilities of generative AI in language models, Chat-GPT offers innovative solutions for project management challenges. This paper also discusses the benefits and challenges associated with integrating Chat-GPT into project management workflows. Through this exploration, we shed light on how Chat-GPT can enhance project management practices, streamline communication processes, and improve overall project throughput and outcomes. The research findings provide valuable insights for practitioners and organizations seeking to harness the power of generative AI in applied project management.

Keywords : LLM; ChatGPT; SPM; Project Management; Artificial Intelligence, Google Bard

1 INTRODUCTION

1.1 Background and Motivation

In today's fast-paced and complex business environment, project management plays a critical role in ensuring the successful execution of projects. Project management is the application of processes, methods, skills, knowledge, experience and a higher order critical thinking capabilities to achieve specific project objectives according to the project acceptance criteria within agreed parameters. Project management has final deliverables that are constrained to a finite timescale and budget[1].It encompasses various aspects, including project planning, scheduling, resource management, risk assessment, and communication. With the emergence of advanced technologies in recent years, there has been a growing interest in leveraging artificial intelligence (AI) and natural language processing (NLP) models to enhance project management practices. Chat-GPT, a state-of-the-art large language model(LLM) developed by OpenAI, has shown tremendous potential in transforming how project management tasks are approached and executed. The integration of Chat-GPT in project management introduces new possibilities for improving communication, decision-making, risk management, and resource allocation. By leveraging the capabilities of language models, Chat-GPT can assist project managers and teams in various ways, such as generating automated project updates, providing real-time collaboration and feedback, identifying and assessing project risks, optimizing resource allocation, and supporting decision-making processes. However, it is essential to evaluate the worth and effectiveness of incorporating Chat-GPT into project management workflows. While Chat-GPT can bring several advantages, there are also considerations regarding its limitations and potential challenges, such as ethical and privacy concerns, the need for fine-tuning and customization, and the potential for biased outputs. Furthermore, the effects of language models like Chat-GPT in software project management are particularly notable. Software projects often involve complex requirements, dynamic changes, and extensive communication among stakeholders. Chat-GPT can facilitate better communication between developers, project managers, and clients by automating routine tasks, providing instant responses to queries, and assisting with software documentation. Additionally, language models can aid in identifying potential risks, analyzing large volumes of software-related data, and supporting decision-making processes in software project management.

1.2. Research Objectives

The primary objectives of this research are to explore the applications of Chat-GPT in applied project management and to assess the potential benefits and challenges associated with its integration into project management workflows. By addressing these objectives, this research aims to contribute to the knowledge and understanding of how Chat-GPT can enhance project management practices and improve overall project outcomes. Specifically, the research objectives are as follows:

1. Identify and examine the potential uses of Chat-GPT in software project management: This research aims to identify the specific areas within project management practices where Chat-GPT can be effectively applied. This includes exploring its applications in communication, risk management, resource allocation, and decision-making processes and more.

2. Assess the impact of Chat-GPT on project management: By analyzing the effects of integrating Chat-GPT into project management workflows, this research aims to evaluate how it can improve project management effectiveness, throughput and output. This includes investigating the potential for enhanced communication, streamlined processes, and improved decision-making outcomes.

3. Evaluate the benefits and challenges of integrating Chat-GPT in project management: The research seeks to identify and discuss the advantages and disadvantages associated with the use of Chat-GPT in project management. This includes examining the potential benefits, such as increased efficiency, enhanced collaboration, and improved project outcomes, as well as the challenges, such as ethical considerations, privacy concerns, and limitations of the technology.

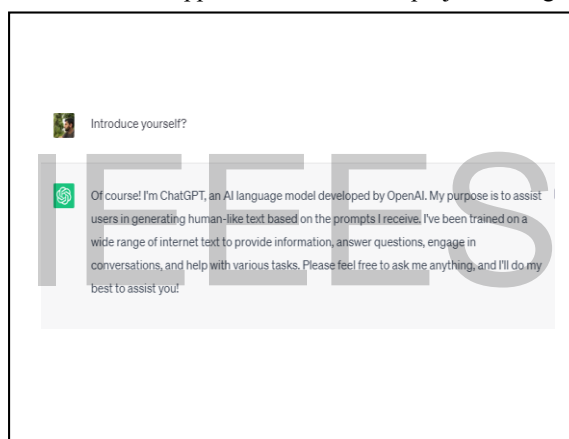
4. Provide insights and recommendations for practitioners in the project management industry: The research aims to offer practical insights and recommendations for project managers and professionals in the project management industry. This includes guidance on how to effectively integrate Chat-GPT into project management practices, potential use cases, and considerations for ethical and responsible AI implementation.

This paper is structured as : Section 1 is the introduction, Section 2 provides a general overview about ChatGPT, large Language Models and Generative Networks. Section 3, provides an overview on how to incorporate LLMs into software project management and finally section 4 concludes this paper.

2 OVERVIEW OF LLMs AND CHATGPT

2.1 Introduction to ChatGPT

Prior to delving into the details of ChatGPT's application in software project management, we queried ChatGPT to explain itself:



Chat-GPT is a state-of-the-art language model developed by OpenAI. It builds upon the success of its predecessor, GPT-3(which stands for Generative Pretrained Transformer), and represents a significant advancement in natural language processing and generation. It is an advanced language model that utilizes deep learning techniques to produce text responses that closely resemble human-generated content. It was introduced by OpenAI on November 30, 2022, as the newest version of their large language models, specifically designed to engage in intelligent and interactive conversations. ChatGPT belongs to the broader family of Generative pre-trained Transformer[2] (GPT) models developed by OpenAI, known for their ability to generate high-quality text based on pre-training and fine-tuning processes. Current version of ChatGPT distinguishes itself from previous models, particularly GPT-3, through its specific focus on certain natural language processing (NLP) tasks. While GPT-3 excels at a wide array of NLP applications like language translation, text summarization, question answering, and creative writing, including poetry and fiction generation, ChatGPT is purpose-built for an array of use cases. It leverages the foundation of GPT-3 but with a specialization in generating responses for dialogues or conversations, explaining complex subjects or concepts, and even assisting in generating or troubleshooting code. This expanded range of applications makes ChatGPT more versatile compared to GPT-3. Additionally, it is worth mentioning that OpenAI[3] has released the GPT-4, a large-scale multimodal model that accepts both image and text inputs and produces text outputs. Although GPT-4 may not surpass human-level performance in many real-world scenarios as of now, it has demonstrated human-level performance on various professional and academic benchmarks.

2.2. Key Features and Capabilities

ChatGPT exhibits several key features and capabilities that contribute to its effectiveness in generating text-based responses. Building upon the advancements of GPT-3 and its specialization in dialogue and conversation, ChatGPT has following notable capabilities:

1. **Natural Language Generation and Natural Language Understanding:** ChatGPT leverages its deep learning architecture to generate human-like text responses. It has been trained on vast amounts of diverse textual data, enabling it to understand and mimic natural language patterns, styles, and nuances.
2. **Context Awareness and Keep Memory of Previous Context:** ChatGPT demonstrates an understanding of context within a conversation. It can maintain coherence and relevance in its responses by taking into account preceding prompts and maintaining a consistent dialogue flow.
3. **Conceptual Explanation:** One of ChatGPT strengths lies in its ability to explain complex subjects or concepts. It can break down intricate ideas into more accessible explanations, facilitating knowledge transfer and understanding.
4. **Code Generation and Debugging:** ChatGPT can assist in generating new code or identifying and fixing errors in existing code. This capability makes it a valuable tool for developers, enabling them to streamline the coding process and troubleshoot issues more efficiently.
5. **Multidimensional Inputs (GPT-4):** As mentioned in the previous context, OpenAI's GPT-4 model can accept both image and text inputs and generate text-based outputs. This multimodal capability expands the potential applications of ChatGPT to include scenarios that involve both visual and textual information.

3. APPLICATIONS OF CHATGPT IN SOFTWARE PROJECT MANAGEMENT

3.1 Enhanced Communication and Collaboration

The term communication comes from the Latin word 'communicare', which means 'to make common'. So, it is defined as "the process of creating meaning between two or more people through the expression and interpretation of messages" [4]. When it comes to project Management communication especially effective communication plays a crucial role in its success. Effective communication channels are essential for facilitating the exchange of information, ideas, and instructions among various project stakeholders such as team members, clients, suppliers, and other relevant parties. Communication plays a vital role in conveying the areas of cost, scope, time, and quality, which are interconnected in project management. It serves as a means to integrate these elements and achieve a high-quality end product. Communication can be viewed as a foundational function that brings together cost, scope, and time, ensuring their cohesive interplay and contributing to the successful delivery of the project[5]. Effective collaboration and the integration of information and knowledge among project team members are crucial for achieving project objectives. In order to facilitate this process, team members should have an understanding about the components of communication. At its core, communication has the following parts: sender, receiver, medium and feedback. In projects, particularly those involving remote team members or isolated work environments like software development and construction projects, feedback assumes a critical role. It becomes especially important in scenarios where team members and head offices are physically separated. Feedback serves as a valuable mechanism for exchanging information, validating progress, and providing constructive input[6]. Conversely, ineffective communication, whether due to miscommunication or lack of communication, can have detrimental effects on the progress and success of a project. It can result in a range of negative outcomes, including: Inadequate Task Definition, Critical Process Failures, Undefined Responsibilities, Failure to Achieve Objectives, Scope Creep (working beyond defined project scope)

3.1.1 Enhancing Project Communication with Chat-GPT

To fully understand the integration of Chat-GPT in project communication, it is essential to grasp the concept of Project Communications Management. This encompasses a set of processes that ensure the timely and appropriate planning, collection, creation, distribution, storage, retrieval, management, control, monitoring, and disposition of project information[7]. Project managers dedicate a significant portion of their time to communicating with team members and various project stakeholders, both internal and external to the organization. The processes of Project Communications Management interact with each other and with processes from other Knowledge Areas. The key processes include:

1. **Plan Communication Management:** Develop an appropriate approach and plan for project communications based on stakeholders' information needs, requirements, and available organizational assets.
2. **Manage Communications:** Focused on creating, collecting, distributing, storing, retrieving, and ultimately disposing of project information in alignment with the communications management plan.
3. **Control Communications:** Monitoring and controlling communications throughout the project life cycle to ensure that the information needs of project stakeholders are met.

Integrating ChatGPT or similar language models into project communication can be beneficial in specific areas. While human involvement and decision-making remain crucial in addressing unknown challenges, ChatGPT can provide valuable support in various aspects of project communication. The key benefits include:

1. **Intelligent Responses and Support:** ChatGPT can act as a virtual assistant, offering intelligent responses and guidance to project team members and stakeholders. It can help answer common queries, provide clarification on project-related matters, and offer assistance with processes and procedures.
2. **Real-time Communication:** ChatGPT enables real-time communication through instant messaging or chat interfaces, promoting efficient collaboration and information exchange among team members, particularly in remote or separated project teams.

3. **Language Translation:** In multicultural and global projects, ChatGPT can facilitate effective communication by assisting with language translation. It helps bridge language barriers, enabling seamless collaboration and understanding among team members who speak different languages.
4. **Documentation Assistance:** ChatGPT can support project documentation tasks by generating drafts, summarizing meeting minutes, and aiding in report writing. This feature saves time, enhances accuracy, and improves the quality of project documentation.
5. **Knowledge Sharing and Retrieval:** ChatGPT can serve as a knowledge repository, storing and retrieving project-related information. Team members can access relevant project documents, previous discussions, and best practices through the chat interface, promoting knowledge sharing and reducing information silos.
6. **Task Reminders:** By integrating ChatGPT with project management tools, team members can receive timely notifications, reminders, and updates on project tasks and deadlines. This functionality helps keep individuals organized, focused, and on track.
7. **Decision Support:** ChatGPT can provide insights and analysis based on project data, offering suggestions, conducting data analysis, and presenting different perspectives for decision-making processes. It assists project stakeholders in evaluating options and making informed decisions.

While integrating ChatGPT into project communication can offer notable advantages, it is crucial to be mindful of its inherent limitations. One important consideration is the potential for generating incorrect or misleading information if the context provided to ChatGPT is unclear or insufficient. Furthermore, the lack of verification mechanisms can make it challenging to validate the accuracy or timeliness of the information generated. Additionally, certain aspects of communication, such as persuasion, negotiations, providing encouragement or reassurance, and resolving conflicts, require human instincts and emotional intelligence that ChatGPT lacks.

3.2 Requirements Gathering

3.2.1 Structured Requirement Documentation

When prompted, ChatGPT can generate a concise and well-structured requirement which might be generic assuming if the prompter did not prompt all the details of the project or the prompt just could not handle that many keywords. In either case the generated documentation can act as a useful and concise starting point for brainstorming with the clients. Moreover, for specific documentation, ChatGPT can transform unstructured conversations into structured documentation. By summarizing and organizing the gathered information, ChatGPT can create well-structured requirement specifications or user stories that can be easily shared and reviewed by stakeholders, development teams, and other project participants.

[Prompt] list down the requirements for a basic chat application. make sure to make it according to a requirements file which will be used by a project manager. This chat application is special as it also lets the users store files within the app-provided cloud storage.

[Response]: Sure! Here's an example of a requirements file for a basic chat application with integrated cloud storage:

Chat Application Requirements

User Management

1. The application should allow users to register an account with a unique username and password.
2. Users should be able to log in securely using their credentials.
3. The system should support user roles such as administrators and regular users.

Chat Functionality

1. Users should be able to create chat rooms and join existing ones.
2. The application should provide real-time messaging capabilities within chat rooms.
3. Users should be able to send text messages, emojis, and multimedia files (images, videos, audio files) within chat rooms.
4. The chat system should support private messaging between individual users.
5. Users should have the ability to edit or delete their own messages within a specified time window.
6. The application should provide notifications for new messages and mentions in chat rooms.

Cloud Storage

1. The chat application should include a cloud storage feature to allow users to store and retrieve files.
2. Users should be able to upload files of various formats to the cloud storage.
3. The system should support organizing files into folders for better organization.
4. Users should have the ability to preview files (e.g., images, documents) stored in the cloud storage.
5. The application should allow users to download and delete their files from the cloud storage.
6. The cloud storage should have sufficient capacity and scalability to accommodate multiple users and their files.

Security and Privacy

1. The application should enforce secure communication using encryption protocols.
2. User passwords should be securely hashed and stored in the database.
3. Access to chat rooms and files should be restricted based on user permissions and roles.
4. The cloud storage should have proper access controls to protect user files from unauthorized access.

User Interface

1. The chat application should have an intuitive and user-friendly interface for easy navigation and interaction.
2. The user interface should support different screen sizes and be responsive for mobile devices.
3. The application should provide search functionality to find specific chat rooms or messages.
4. The cloud storage interface should allow users to browse, search, and manage their files effectively.
5. The user interface should display real-time updates of new messages and file uploads.

Performance and Scalability

1. The application should be able to handle concurrent users and provide a responsive user experience.
2. The system should have mechanisms in place to handle potential spikes in user activity.
3. The cloud storage should be scalable to accommodate increasing file storage demands.
4. The application should optimize data retrieval and minimize latency for efficient messaging and file operations.

Integration and Compatibility

1. The chat application should be compatible with popular web browsers (e.g., Chrome, Firefox, Safari) and support the latest versions.
2. The application should have APIs or integration capabilities to allow integration with other systems or services if required.

These requirements serve as a starting point for developing a basic chat application with integrated cloud storage. The project manager can further refine and expand upon these requirements based on specific business needs and user expectations. Notice that the response was generated in markdown format automatically without the need to explicitly ask for it. We can safely assume here that ChatGPT evaluated the request and considered the possibility that the response would be used in a markdown file with source code of the project. Furthermore, in a more analytical perspective, we can see that the list generated by ChatGPT covers all the generic functionalities as well as the special requirements put forward by the prompter. One thing worth noting is that, not only did it provide a basic layout of functionalities which should be present in the chat application as requested but also considered the possibility of security risks and pointed out the generic way of dealing with those issues. Apart from that it also considered the potency of the app and put them under the “Performance & Scalability” header. While considering the possibilities of the chat application, since we did not specify that it is going to be a mobile app therefore it suggested that we should account for its compatibility in different browsers and that it should have API endpoints available to make it easier to integrate it into other systems and platforms such as phones.

3.2.2 Brainstorming and Idea Generation

For a project manager who is brainstorming solo, it can act as a creative partner which can provide insights to thought processes and simulate various ideas as well. One of the examples are given below:

[Prompt] Certain web applications can be used to communicate with a wide variety of audience via broadcast or end-to-end chats, while also providing features such as privacy options which protects the user from unwanted contacts. These types of applications can also be used for marketing purposes or allow us to create a forum for a specific topic which anyone can join, for which privacy of joining options set by the admin of those forums so that unwanted contacts are not allowed to join. Can you provide examples of similar projects or products that have successfully addressed similar needs?

[Response] Certainly! Here are a few examples of projects or products that have successfully addressed similar needs of communicating with a wide variety of audiences, providing privacy options, and supporting marketing or forum creation:

1. Slack:
3. Telegram:
4. WhatsApp Business:
5. Facebook Groups:

These examples showcase how various projects and products have successfully addressed the needs of communicating with diverse audiences, incorporating privacy options, and supporting marketing or forum creation. Each platform has its own unique set of features and capabilities, catering to different use cases and requirements. As we can see from the example that the response was concise enough to be considered a “quick look-up” type of response but was also to the point as required by the prompt.

3.3. Managing Constraints

The term "constraint" refers to any imposed limitation or restriction. In the context of software development, constraints are predetermined factors established at the start of a project, including elements such as time, budget, or available resources[8]. Effectively managing these constraints requires adopting a management framework, such as the Theory of Constraints (TOC), which provides a systematic approach to optimize project performance and address limitations. The concept of the imaginary triangle with three sides, namely Cost/Resource, Schedule/Time, and Scope/Quality, is commonly referred to as the Project Management Triangle or the Triple Constraint. It illustrates the interdependence and trade-offs among these factors in project management[9]. The Project Management Triangle states that these three factors are interconnected, and any change in one factor will have an impact on the other two.

By proactively considering constraints and integrating ChatGPT into project management processes, organizations can greatly enhance their chances of achieving project success. When project teams leverage ChatGPT's capabilities, they can anticipate and manage constraints more efficiently.

1. Exchange Information: The ability to communicate and exchange information in real-time fosters a collaborative environment where team members can discuss and address potential constraints as they arise.
2. Identify interdependencies: By harnessing the power of ChatGPT, project teams can identify interdependencies between cost/resource, schedule/time, and scope/quality, enabling them to make informed decisions and adjust project plans accordingly.
3. Optimal Resource Allocation: By analyzing project data and considering different scenarios, ChatGPT can suggest optimal resource allocations that align with project constraints. This empowers project teams to optimize resource utilization, identify potential bottlenecks, and make timely adjustments to keep the project on track.
4. Improving from past Knowledge: facilitates continuous improvement by enabling organizations to learn from previous projects and apply lessons learned to future endeavors. By storing and retrieving project-related information, including documentation, discussions, and best practices, ChatGPT serves as a knowledge repository.

3.4. Aiding in Project Estimation

Estimation in project management has often been referred to as a "black art" due to its perceived subjectivity[10]. On the surface, it may appear that estimation is a highly subjective process, as individuals can have different perspectives on the time required to complete a task. This variability in estimates becomes evident when multiple people are asked to estimate the duration of a task, resulting in widely differing answers. The first step for a sound estimation is to create list of possible tasks that have to be completed in order to a successful project also known as WBS (Work Breakdown Structure). Once the Work Breakdown Structure (WBS) is established, the project team needs to estimate the effort required for each task. The most accurate estimates are often based on prior experience and historical data. It is essential for team members to review the results of previous projects and analyze how long similar tasks took to complete. Estimation is an iterative process that requires continuous monitoring, adjustment, and learning from past experiences. It is through this process that project teams can refine their understanding of tasks, uncover risks, and adapt their plans accordingly. The goal is not to achieve perfect accuracy, but rather to make informed decisions and manage expectations based on the best available information at any given time. So, now the question that arises is whether we can integrate ChatGPT or similar applications to assist in project estimation? In order to prove our point, we queried ChatGPT with following context:

[Prompt] "Let's say we are interested in developing a chat application. Where a user can register via his phone number. An OTP is sent to his phone number for verification. After that, he could enter his name, profile picture and then be able to create chats with other users. We also want to add an option of channels and groups. Users can subscribe to channels and groups. the difference between channels and groups is that users can join groups but groups have admin and admins have privileges to create an add a user, remove user and make someone else admin. and as well a channel can have sub channels. On the other hand, groups don't have sub groups and admin have same privileges as of channels. Create a WBS? "

[Response] Project Initiation: 1 week

Define project goals and objectives.

Conduct a feasibility study and define project scope.

1. *User Registration and Verification: 8 hours*

Develop user registration functionality.

Implement OTP verification process.

2. *User Profile Management: 8 hours*

Allow users to enter their name and profile picture.

Implement profile management functionality.

3. *Chat Creation and Management: 3 days*

Develop the ability for users to create and manage individual chats.

Implement real-time messaging functionality.

4. Channel Management: 3 days

Design and develop the ability for users to create and subscribe to channels.

Implement channel management features such as channel creation, subscription, and messaging.

5. Group Management: 3 days

Develop the ability for users to create and join groups.

Implement admin privileges for group management, including user addition, removal, and admin designation.

6. Subchannels and Subgroups: 3 days

Design and implement the functionality for channels to have subchannels.

Develop subgroup management features for groups with admin privileges.

7. User Interface and Experience: 8 hours

Design and develop an intuitive and user-friendly interface.

Ensure smooth navigation and seamless user interactions.

8. Testing and Quality Assurance: 2.5 days

Conduct thorough testing to identify and fix any bugs or issues.

Ensure the application functions correctly across different devices and platforms.

9. Deployment and Maintenance: 1.5 day

Deploy the application

In the above, response we have added time per WBS task. Now with this response and time we again queried ChatGPT and this time we are interested in finding the cost and time required for the project. Cost of a project may depend on various factors so let's assume an ideal case where we have:

[Prompt] we have following resources and requirements 1 Android developer 1 iOS developer 2 backend developers Developers with 2 years of experience Developers work 8 hours a day, 5 days a week. Average Salary of Mobile Application Developers ~ \$82k/year. [11]

With this context, we queried ChatGPT and we get following response:

[Reponse] Now, let's calculate the total effort for the project by summing up the effort for each task:

Total Effort = 40 hours + 8 hours + 8 hours + 120 hours + 120 hours + 120 hours + 8 hours + 100 hours + 60 hours = 584 hours

Given that you have 1 Android developer, 1 iOS developer, and 2 backend developers, all working 8 hours a day, 5 days a week, we can calculate the project duration:

*Total project duration = Total effort / (Number of developers * Hours per day)*

*Total project duration = 584 hours / (1 + 1 + 2) * 8 hours/day = 584 hours / 32 hours/day = 18.25 days*

Next, let's estimate the project cost based on the average salary of mobile application developers:

Average Salary per Developer = \$82,000/year

*Average Salary per Developer per Hour = \$82,000/year / (52 weeks * 5 days/week * 8 hours/day) = \$62.50/hour*

*Total Project Cost = Total effort * Hourly Rate*

*Total Project Cost = 584 hours * \$62.50/hour = \$36,500*

Based on the estimation, the project is expected to take approximately 18.25 days (or about 3.5 weeks) and would cost \$36,500.

So by looking at above calculations we could have an overview of what this project estimation would look like. However, it's important to note that these estimates are based on the provided information and assumptions, and actual project timelines and costs may vary depending on various factors such as the complexity of the implementation, unforeseen challenges, and individual developer productivity.

3.5. Tracking and Troubleshooting

3.5.1. Issue Tracking

Issue tracking is usually done through an issue tracker software which enables the project manager to manage and address issues. Issues are any roadblocks or unintended impacts that directly affect your project's timeline and or performance. An issue tracking system can help you identify, prioritize, assign, monitor, and resolve issues in a timely and efficient manner. Since the issue tracker software includes a knowledge base of all the issues that were raised, addressed or postponed, ChatGPT, when integrated to the system or given access to the knowledge base of the system, may be able to use that knowledge to assist in identifying and prioritizing the list of hot issues which need to be immediately addressed, or provide a list of actions that can be taken to resolve a certain issue or generate suggestion as to whom should a certain issue be assigned to. As far as monitoring is concerned, It is only needed for notifying concerned personnel about a selected list of events. This can be achieved through automation and does not necessarily require a Chabot. But It can be helpful in summarizing the event that has occurred

according to the interests of the concerned. One of the most frequent uses of ChatGPT is in troubleshooting. People often state their problems to it and it generates some suggestions on how one should address the problem.

[Prompt] How would you handle a situation where you discover a major software bug or security vulnerability after the product has been launched or delivered to the client?

[Response] This is a challenging situation that requires quick and effective action. Here are some possible steps I would take to handle it:

- *Assess the impact and urgency of the bug or vulnerability ...*
- *Inform the client and stakeholders: ...*
- *Mobilize the team and resources ...*
- *Implement a solution and test it ...*
- *Deploy the solution and verify it ...*
- *Follow up with the client and stakeholders ...*

In the above response to a general project management problem, ChatGPT laid out several actions that can be taken to address the situation. We received several options since we did not specify our limitations. For example, we can assume that the third point that it suggested is not feasible for us (due to any arbitrary reason) and ask ChatGPT again what it suggests to us.

[Prompt] I cannot mobilize my team and resources since i do not have that much manpower. The developers I have are already working on important things and they will not be able to take some time out for this issue. What do you suggest I do now?

[Response] This is a difficult situation that requires some trade-offs and compromises. Here are some possible options I would consider:

- *Re-prioritize the current tasks and allocate some developers to the issue: ...*
- *Hire or outsource some additional developers to work on the issue: ...*
- *Negotiate with the client and stakeholders for more time or resources: ...*

One more notable aspect is that negotiation and follow-up with the clients is found to be the very last resort if we were to sequentially assess every option. We can safely assume that ChatGPT accounted for priority of the methods as well while suggesting us actions which were needed to troubleshoot the issue at hand.

3.6. Troubleshooting and Debugging Code

One of the most important issues that programmers often encounter when solving problems, regardless of their size or complexity, is the approach they choose. At times, we confront challenges that initially appear too complex to grasp and even seem insurmountable when attempting to implement solutions. However, upon gaining a clear understanding of the problem, alternative, more effective pathways often become apparent. These pathways can lead to solutions within the designated timeframe and budget. When ChatGPT is brought into such situations, all you need to provide is the context and your preferred language. It can then offer multiple responses for each query, unveiling diverse ways to attain the desired goal. This dynamic exploration opens up numerous potential routes to reach the solution. Consider following real scenario:

[Prompt] DemoApp[5157:180474] { FIRAuthErrorUserInfoNameKey = "ERROR_KEYCHAIN_ERROR"; NSLocalizedDescription = "An error occurred when accessing the keychain. }... EXPLAIN AND HOW CAN I RESOLVE THIS ERROR?

This is a real case, where we were working on a chatting application, we had configured everything as mentioned in the official documentation of Firebase but still unable to reach a solution so we queried ChatGPT to look for its probable causes and the response was quite significant. It responded like a programmer with many years of experience that it showed us solutions(Plural) to this particular problem as shown in following:

[Reponse] Check Keychain Entitlements: Check Keychain Access Groups:Check for Relevant Documentation: Test on Different Devices/Environments: Update Firebase SDK: Contact Firebase Support:

Software projects often consist of extensive lines of code, and the presence of code duplication is a common issue, especially for less experienced programmers. This tendency arises from our inclination to copy and paste sections of code, which can lead to a codebase filled

with redundant elements. In situations where project resources are limited, and the project itself is substantial and fraught with challenges, programmers might forsake sound programming practices. Instead, they might resort to writing solutions that seem to address immediate issues. While this approach might yield a functional solution, it often comes at the cost of neglecting proper programming practices and ignoring the crucial practice of continuous refactoring. Regrettably, the consequence of this behavior is that the final product may suffer from poor performance and could even fail to execute efficiently on client machines. This outcome results from the failure to adhere to established best practices and the lack of regular code optimization.

In such scenarios, where programmers are pressured to focus solely on writing code rather than embracing good programming principles and consistently refactoring, the assistance of ChatGPT can be invaluable. ChatGPT can provide guidance on coding best practices, suggest ways to refactor and optimize existing code, and help programmers identify more efficient solutions to their challenges. By incorporating ChatGPT into the development process, programmers can receive real-time support that encourages the implementation of better coding practices, ultimately leading to a more efficient and maintainable end product.

4. CONCLUSION

Through a thorough exploration of Chat-GPT's capabilities, this research identified numerous areas where the model can offer substantial benefits to project management. From enhancing communication and collaboration through real-time interactions and language translation to aiding in requirement gathering, brainstorming, and issue tracking, Chat-GPT proves to be a versatile tool for project teams. Its ability to generate structured documentation, offer intelligent responses, and provide decision support enhances project managers' capabilities to address constraints and facilitate effective estimation. The application of Chat-GPT in software project management brings forth a set of advantages that must be balanced with potential limitations. While Chat-GPT can significantly improve communication, automate routine tasks, and provide valuable insights, it should be recognized that it lacks emotional intelligence and may generate incorrect or misleading information if not guided correctly. Ethical considerations, privacy concerns, and the need for human expertise in certain domains must also be taken into account. Furthermore, this paper demonstrated Chat-GPT's role in project estimation through the creation of a Work Breakdown Structure (WBS) and the subsequent calculation of project duration and cost. While the provided estimates serve as a useful guideline, project managers should understand that variations in complexity, unforeseen challenges, and individual productivity levels can impact actual project timelines and costs.

REFERENCES

1. APM Body of Knowledge. / Murray-Webster, R.D., Darren., *APM Body of Knowledge 7th edition*. 7th ed ed. 2019, Princes Risborough: Association for Project Management. 230.
2. Alec Radford, K.N., Tim Salimans, Ilya Sutskever, *Improving Language Understanding by Generative Pr*. 2018. 12.
3. OpenAI* , *GPT-4 Technical Report*. 2023.
4. Sandra Cleary, M.H., et.al, *The Communication Handbook*. 2007, Republic of South Africa: Juta & Co.
5. Zulch, B., *Communication: The Foundation of Project Management*. Procedia Technology, 2014: p. 1000-1009.
6. Talukhaba, A.M., Thomas & Miruka, Collins, *Indicators of effective communication in remote projects*. International Journal of Project Organization and Management, 2010.
7. *A guide to the project management body of knowledge (PMBOK® guide)*. -- Fifth edition. 2013, 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA: Project Management Institute, Inc.
8. Antoine Asseman, E.R., Heba Dawoud, Mariam Salim, Nada Ashqar, Nasser Alkhuzaei, Rana Rihan, Souad Mecheter, Abdelaziz Bouras, *Software Project Management: Theory of Constraints, Risk Management, and Performance Evaluation*. THE JOURNAL OF MODERN PROJECT MANAGEMENT, pp:9-21, 2014.
9. Eliyahu M. Goldratt, J.C., *The Goal: A Process of Ongoing Improvement*. 1992: North River Press, 1992.
10. Andrew Stellman, J.G., *Applied Software Project Management*. November 2005, O'Reilly Media, Inc. 1005 Gravenstein Highway North Sebastopol, CA 95472.: O'Reilly.
11. Glassdoor. Available from: <https://www.glassdoor.com/>.