



Prompt Patterns for Agile Software Project Managers: First Results

Kari Sainio[✉] , Pekka Abrahamsson, and Tero Ahtee

University of Tampere, 33100 Tampere, Finland
kari.sainio@gmail.com

Abstract. In the evolving field of Agile Project Management (APM), the role of the project manager is in transition. This paper identifies common ‘pain points’ in APM through a literature review and constructs a theoretical model to address them. The study introduces ‘Prompt Engineering’ as a novel approach to leverage artificial intelligence (AI), specifically ChatGPT, for mitigating these challenges. Empirical research evaluates ChatGPT’s capabilities and reliability in managing various project tasks using engineered prompts. The findings suggest that while ChatGPT cannot fully replace human project managers, it excels in assisting, guiding, and automating specific tasks when guided by well-crafted prompts. As an outcome, prompt engineering patterns for project managers is proposed to facilitate the application of AI in agile settings. In this paper, we introduce patterns for requirements management, stakeholder and management teams and role clarification. The paper concludes that ChatGPT’s knowledge is generally reliable but emphasizes the need for expert evaluation in critical areas.

Keywords: Agile Project Management · Pain Points · Artificial Intelligence · LLM · ChatGPT · Prompt Engineering · Patterns

1 Introduction

Project management in the IT sector faces a myriad of challenges, particularly within the realm of Agile Project Management (APM) [1]. APM, an empirically driven approach, aims to adapt to environmental changes to ensure project success [2]. However, it confronts multi-level challenges ranging from project scope to team dynamics, individual performance, and task management [3, 4]. These challenges, often termed as ‘pain points,’ necessitate strategic and adaptive practices for successful project execution [5].

Moreover, challenges can be scope creep, where projects expand beyond their original objectives, causing time and budget overruns. Resource management can be another challenge, with unexpected changes in personnel or material resources leading to delays. Additionally, unclear communication among team members can lead to confusion and inefficiencies. Constant shifts in the business or regulatory landscape also add to the complexity, necessitating frequent adjustments in project direction. Lastly, stakeholder management can be difficult, as varying interests and expectations may conflict with project goals. These kinds of challenges can be called pain points which are examples

that must be paid attention to strategic and adaptive project management practices to ensure success.

In recent years, Artificial Intelligence (AI) has evolved to include systems proficient in natural language processing [6]. Conversational AI (CoAI) bots like E-Commerce Customer Service Bots and Amazon Echo Alexa have gained widespread use [6]. Advanced AI systems like ChatGPT have emerged, capable of conducting dialogues and providing solutions to various user queries [6]. Generative AI (GenAI) models can produce high-quality text and other content based on their training data [6]. These AI technologies offer promising avenues for automating or assisting in project management tasks.

The increasing adoption of APM in IT related projects demands a high level of discipline and skill from both the project organization and the project manager [7]. Given the advancements in AI techniques like machine learning and machine reasoning [8], there's a growing interest in exploring AI's role in automating or delegating specific project management tasks [9].

This study aims to investigate the applicability of AI, particularly GenAI models like ChatGPT, in managing the challenges and pain points in APM. The research questions guiding this study are:

- RQ1: What are the typical pain points in agile projects?
- RQ2: How can GenAI guide the mitigation of these pain points?

By addressing these questions, this study endeavors to provide a comprehensive understanding of AI's potential in enhancing APM practices.

2 Pain Points for Agile Projects

In the realm of software engineering, the adoption and scaling of agile methodologies are fraught with challenges that are both intricate and context sensitive. Patel et al. [10] underscore that team members accustomed to structured methodologies like Waterfall often resist transitioning to Agile. This resistance is compounded by a general lack of understanding of Agile principles among team members and insufficient involvement from top management. Nuottila et al. [11] extend the discourse to the public sector, identifying additional challenges such as documentation, stakeholder communication, and legislative constraints. The complexity is further exacerbated when different Agile methodologies like Scrum, XP, and Lean are mixed [12]. While the Agile paradigm has been widely adopted, certain areas like governance, business engagement, and IT transformation remain under-researched [13]. Dikert et al. [12] enumerate challenges in scaling agile, including change resistance at organizational levels, misunderstandings of Agile concepts, and issues with work estimation.

The advent of remote work, accelerated by the Covid-19 pandemic, has introduced its own set of challenges such as fewer organic interactions and meeting overload [14]. Reunamäki et al. suggest mitigations like smaller sub-teams and increased leader presence to address these remote work challenges [14]. Paasivaara et al. [15] discuss challenges in global companies adopting Agile, such as technical debt and lack of a common Agile framework. Hoda et al. [4] categorize challenges at project, team, individual, and

task levels, emphasizing issues like delayed requirements and senior management sponsorship. Sithambaram et al. employ a grounded theory approach to divide challenges into organizational, people, process, and technical factors [16]. Shameem et al. [17] extend the classification into management, team, technology, and process in the context of distributed software development. In summary, the challenges in APM are multifaceted and often interlinked, requiring a nuanced understanding and tailored solutions for effective implementation.

A distinct model addressing these pain points is introduced, aiming to provide solutions for common issues in agile endeavors. To devise this model, challenges were categorized. Although numerous classifications exist on the subject, this study proposes one potential arrangement, acknowledging that some challenges might span multiple categories. Five distinct categories were identified, and within each, two predominant challenges were chosen based on their prevalence in academic literature. These documented challenges then informed the suggested solutions to these prominent pain points. To categorize challenges pertaining to pain points, analogous studies on Agile projects were analyzed based on literature research, with their results displayed in Fig. 1. This fishbone has been constructed based on the pain points shown in Table 2 (see further). The classification system somewhat mirrors the one by Sithambaram et al., which includes categories like project, people, process, organizational, and technical [16]. However, this study replaces “organizational” and “technical” with “endurance” and “effort estimation”. In this context, “endurance” predominantly alludes to resistance to change, and the sustained commitment to adhering to Agile principles and practices. At the end of the day “work estimation” and “technical knowledge” correlate with effort estimation as without the knowledge there is no good way to estimate.

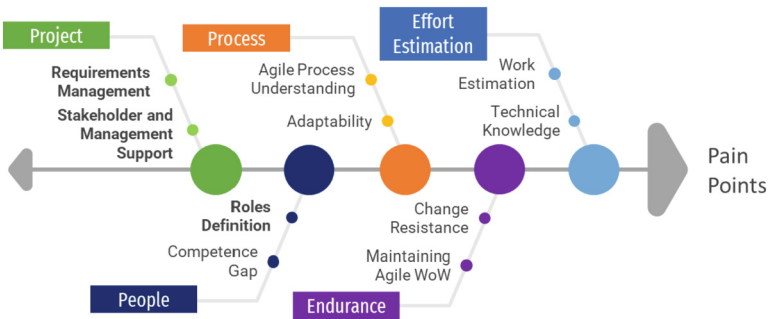


Fig. 1. Fishbone diagram model for APM pain points

So far, we have explored various challenges often faced in the realm of APM. Using insights from existing literature, each section has focused on a specific issue, such as requirements management, stakeholder support, and role definition, among others. For every challenge discussed, we now offer a review of potential solutions that have been suggested by researchers and practitioners. This approach is intended to provide a balanced view of the difficulties involved in APM, along with possible ways to address

them. Our goal is to explore whether an AI can help in using the solution in practice. Table 1. Presents the solutions offered by the literature for each pain point identified.

Table 1. Solutions from the literature for the identified pain points

Pain point	Description	
Requirements Management	Effective communication and requirement traceability are key in ARCM, supported by tools like Jira	[18]
Stakeholder and Management Support	Management's commitment to agile values and principles, along with executive sponsorship, is crucial for project success	[19]
Role Definition	Clear role definitions facilitate self-organization, but overburdening specific roles can be a challenge. Scrum Masters can help remove blockages	[20]
Redundancy	Talent management and continuous learning are essential for addressing redundancy and competence gaps	[13]
Agile Process Understanding	Organizations should adopt comprehensive project management tools and methodologies for agile success	[19]
Adaptability	Agile coaches facilitate adaptability and self-organization within teams	[21]
Change Resistance	Scrum Masters with strong group management skills and empathy can manage resistance to change	[10]
Maintaining Agile Way of Working	Sustaining an agile approach requires management to have a deep understanding of agile methodologies	[12]
Work Estimation	AI-based Agile Story Point Estimations are considered promising for work estimation	[22]
Technical Knowledge	Technical expertise is crucial for project success, and a SWOT analysis can be beneficial for skill evaluation	[23]

3 Research Design

In this section, the research process and methodology get delineated as shown in Fig. 2. The initial phase of the study introduces various agile frameworks. Illustrations of these frameworks in larger, scaled-up applications in substantial agile projects make up part

of this exploration. These frameworks receive classifications into small-scale and large-scale. The small-scale group includes Scrum, XP, Kanban, and Lean Software Development. On the other hand, the large-scale frameworks include SAFe, LeSS, and DA. The selection of these frameworks derives from insights culled from pertinent literature. The goal remains to provide an encompassing introduction and guidance on these frameworks' use.

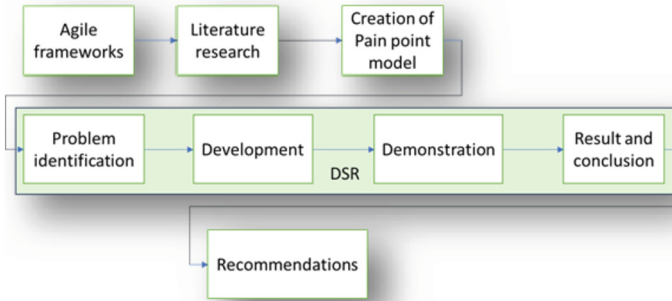


Fig. 2. Research process (Design Science Research)

Moving on to the next section, it involves an extensive literature review on the common challenges encountered when adopting and implementing agile methodologies. These challenges are analyzed, categorized, and synthesized into a pain point model shown in Table 1, which is presented as the problem identification for the research.

Design Science Research (DSR) has been employed as methodology for a strategic approach to discover effective GenAI solutions for mitigating these identified pain points (Fig. 2). DSR is an approach to problem-solving that aims to advance human knowledge through the development of innovative artifacts [24]. These artifacts, called **prompt patterns** in this study, are designed to address specific challenges, and enhance their surrounding environment, resulting in an enriched technology and science knowledge base [24]. In DSR research is conducted first identifying the problem, defining the objectives, developing the solution, demonstrating, and evaluating the results [25]. Finally, practical recommendations are made.

3.1 Problem Identification and Objectives Definition

The identified problem is the formulation of appropriate prompts to be used in conjunction with ChatGPT, aimed at easing the pain points commonly associated with Agile projects. Primary objective to assess the possible implementation of ChatGPT as a support mechanism in intelligent APM. Moreover, the objective is to determine distinct prompt patterns that generate precise information. While the creation of prompts can take on many forms, this research does not develop a specific grammar, but instead designs patterns to steer ChatGPT toward providing suitable responses with minimal hallucination, a method supported by White et al. [26]. The prompt patterns used in this study

adopt a similar strategy, abstaining from introducing a unique syntax or language. The aim is to supply relevant keywords that can aid project managers or stakeholders in initiating early dialogues with GenAI, thereby broadening the application of project-specific parameters.

The problem identification in this research hinges on the empirical aspect of design science research, which involves interacting with ChatGPT to evaluate various prompts capable of generating accurate responses for a specific subject, aligning with the method proposed in the White et al. studies [26]. Unlike focusing on prompt patterns to improve code quality, the emphasis is on identifying and assessing prompts that can support Agile projects while mitigating the impact of various challenges.

3.2 Development

In the development phase of the DSR method, prompt patterns (i.e., artifacts) are generated for ChatGPT, designed to assist in mitigating the challenges associated with agile methodologies. A prompt, as defined, is a textual input given by the user, acting as the commencement point for ChatGPT's response generation [27]. A prompt pattern, therefore, is a generalized construct for a specific prompt topic.

The development of these prompt patterns has been involving ChatGPT's web-based interface along with the GPT-4 model. The intention behind this phase of the research was to create a practical and robust means of addressing agile project pain points through specifically crafted prompts. As DSR principle includes several iterations only the final version of the prompts is shown and demonstrated.

According to White et al. a prompt sets the context for the conversation and tells ChatGPT what to focus on and what are the expectations for the output [26]. A specific prompt pattern is implemented to each specified pain point. In conversations with GenAI different types of prompts: explicit, implicit, and creative can be used. Explicit prompts are direct and clear instructions given to the AI model about the specific format or information needed in the output. On the other hand, implicit prompts are less direct and give the AI model more flexibility to interpret the intended result. Creative prompts aim to inspire AI models to produce original, imaginative, or unconventional outputs [28]. An explicit approach for prompt pattern development has been selected. Each prompt pattern developed follows roughly the model introduced by White et al. [26]:

- The name and classification. The name is used to unique identify the prompt pattern. The classification is based on the presented pain point model classification defined in Sect. 2.
- The intent. The purpose of the pattern is conveyed through its intent.
- The motivation behind the pattern is documented, which explains the underlying problem it is intended to solve and why it is important.
- The structure and key ideas. The structure and key ideas describe the fundamental contextual information that the prompt pattern provides to the ChatGPT.
- The demonstration. Providing example patterns helps to demonstrate how the pattern can be applied in agile projects.
- Results are presented as empirical contributions and summarize the advantages and disadvantages of implementing the pattern in practical situations.

Prompts should set the context, define expectations, channel creativity, and reduce ambiguity [27]. Each prompt pattern developed contains the following contextual sentences:

1. ChatGPT (LLM) is asked to act as **[RoleA]** to work (task) in given problem domain.
2. The necessary constraints are presented [**n * C**].
3. ChatGPT (LLM) is *asked to verify common understanding* and clarifications of the constraints.
4. ChatGPT (LLM) is asked to reflect its understanding and provide a solution to **[RoleB]**.
5. Finally, the output format is requested, defined as **[Format]**.

Both RoleA and RoleB represent different and typical project staff roles such as project manager, engineering manager, program director, software developer, and requirements engineer. Constraints can be given as free description or comma separated items. Constraints ($n * C$) can vary from requirements to different objectives according to project needs and are subject to each project.

3.3 Demonstration and Evaluation

The prompt pattern's efficacy is assessed through practical demonstrations. Each prompt is entered using ChatGPT and response is collected for evaluation. An evaluation is done for each prompt and a summary is presented as a contribution. Since the outcomes might be subjective and immeasurable, reference to existing literature is employed to evaluate the effectiveness of each prompt pattern. In demonstrations, hypothetical project management challenges are utilized. These are based on individual experience of author as a project manager. Every demonstration of prompt patterns occurs three times, utilizing the same prompt, ensuring consistency in responses from ChatGPT. During the third issuance of prompts, an additional iteration ensures further consistency.

The research discusses theoretical and practical implications derived from literature findings and observations, offering practical recommendations on how ChatGPT can be employed in agile projects pain points. Ultimately, the research aims to tackle the proposed research questions.

4 Empirical Results

This chapter showcases various prompt patterns and corresponding demonstrations utilized with ChatGPT. Given that ChatGPT can generate extensive responses, only selected portions of these dialogues will be highlighted in the subsequent chapters. Complete, original responses are not presented in this document due to limited space. The result section exhibits each prompt pattern via a sample dialogue with ChatGPT. Each prompt is inputted into ChatGPT thrice, spanning three rounds within a single atomic session, to observe the variations in ChatGPT's responses to identical prompts. These responses form empirical research data. The displayed prompt examples are selected from data. **Empirical Contributions (EC)** and **Primary Empirical Contribution (PEC)** are used to underline the key findings in the prompt responses. Prompt patterns are classified

according to Table 2 so that there are two patterns representing classified pain points. As of June 2023, ChatGPT operates with a maximum token limit of 2048 for a single prompt [29].

Table 2. Prompt Pattern Classification

Classification	Pain point	Prompt pattern name
Project	Requirements management	Requirements engineering pattern
	Stakeholder and management support	Steering group pattern
People	Role definition	Role clarification pattern
	Redundancy	Redundancy analysis pattern
Process	Agile process understanding	Agile process coaching pattern
	Adaptability	Adaptability management pattern
Endurance	Change resistance	Change resistance pattern
	Maintaining agile way of working	Agile way of working pattern
Effort estimation	Work estimation	Work estimation pattern
	Technical knowledge	Technical knowledge

However, the demonstrations employ a specific tool that utilizes a smaller token size verified by tool [30]. Testing has revealed that if the prompt size exceeds this limit, it hampers ChatGPT's ability to respond. It might even cause the model to forget the previously discussed context [31]. Nevertheless, during prompt demonstrations, such behavior was not encountered.

4.1 Requirements Management

Assessing the utility of ChatGPT in assisting project requirements generation, the intention behind this prompt pattern has been to aid in formulating accurate needs that will enable an agile project to address the pain point associated with requirements management and creating requirements. The motivation for this prompt structure has been to assist in establishing well-defined, comprehensible requirements while also facilitating their concretization and traceability. Using this pattern would help the project to create initial requirements and furthermore generally in requirements management.

Requirements Creation Pattern

Contextual statements:

“As a [roleA] your task is to create a requirement specification. The requirement specification must fulfil the following constraints: [c1, c2, c3...].

Ask the necessary questions to fully understand the constraints. Based on my answers create the requirement specification.

Explain your understanding and construct your proposal to me representing the [roleB]. Provide output in [format].”

Prompt Pattern Demonstration: *“As a requirement engineer your task is to create a requirement specification. The requirement specification must fulfill the following constraints: Internet banking system which can handle typical banking transactions to be possible to browse account and make payments. Ask the necessary questions to understand the constraints. Based on the answers create requirement specification. Explain your understanding and construct your proposal to me representing the project manager. Provide your output in table format.”*

For this prompt ChatGPT response in the first prompt test round is: “Sure, I will need a few more details to make sure we’re on the same page. Here are the questions that I’d need answers to...”

EC1: During each iteration, ChatGPT responds to the given prompt by posing contextual questions for further clarity. It also presents preliminary requirement recommendations in a table format, which might differ across rounds. Additionally, the table includes requirements that ChatGPT autonomously suggests.

In the second following prompt clarifications and details to the additional questions are given as freeform. ChatGPT replies and generated refined output of requirements:

“ChatGPT:” Thanks for the information! Based on your responses, here is a revised Table of requirements:

This Table outlines the system’s main functional and non-functional requirements, allowing the developers to understand what they need to build. Additionally, it outlines sources of the requirements, which can be useful for reference and further consultation.

EC2: ChatGPT offers a revised set of requirement specifications in a tabular layout. The specifications are determined by the details provided in the second prompt. The origin of these requirements is only specified in the initial round. Without the presentation of the source requirement, adjustments might be necessary to ensure traceability.

In the third prompt ChatGPT is asked to prioritize the requirements: “Can you prioritize requirements in the table?”. Now the output contains an augmented table with additional column for prioritization (Must have, Should have, Could have, Won’t have).

EC3: In the table, requirements prioritization can be incorporated through an added prompt. This likely holds true for other custom adjustments as well. In the third iteration, ChatGPT introduced a prioritization for the requirements, even though it wasn’t explicitly requested.

In round 3 prompt iteration is demonstrated. Additional requests to the previous prompts can be made and ChatGPT responses to the changes.

EC4: ChatGPT reacts to prompt cycles according to user directions and can grasp supplementary clarifications.

PEC1: ChatGPT is prompted to create requirements based on a given specification pattern. It can seek clarifications, offering recommendations, and producing high-level requirements presented in a tabular format in response to a prompt. However, the content of these requirements may differ depending on the round they were generated in.

4.2 Stakeholder and Management Support

Intention for this prompt pattern is to provide guidance and workflow how project manager could utilize ChatGPT when facing and communicating with stakeholders, project sponsors and provide more transparency and understanding to different project related challenges. Motivation for this pattern is to mitigate pain point where the project stakeholders do not understand the project objectives and how those are implemented in agile way. The prompt pattern is labeled as steering group meant for simulating the guidance and instructions provided by the project steering group.

Steering group pattern

Contextual statements:

“As a [roleA] your task is to fully understand agile principles, business objectives and give feedback about the project.

The project has the following objectives: [o1, o2, o3...].

Ask the necessary questions to fully understand the objectives and to be able to understand the project status and way of working.

Explain your understanding and provide necessary actions for me representing the [roleB]. Provide output in [format].”

Prompt Pattern Demonstration: *“As a project’s steering group member your task is to fully understand agile principles, business objectives and give feedback about the project. The project has the following objectives: We are implanting an ERP integration system to customer using SAP product. Integration is done in agile way, and we expect to release one integration in every two weeks. At the moment we do not get customer to participate demos and would need their feedback. Project cannot start final testing before customer testing. Customer does not have time for the testing but is expecting the release soon. Ask the necessary questions to fully understand the objectives to be able to understand the project status and way of working. Explain your understanding and provide necessary actions for me as the project manager. Provide output in memo format.”*. Memo output is requested in the prompt as an example to give possibility to share the outcome using e.g., email. Its format has not been defined.

The first-round response of ChatGPT contains: “... To further understand the situation and propose effective solutions, I need additional information on the following:

1. Customer Engagement: What are the current methods of customer engagement and how frequently are we reaching out to them for demos and feedback?
2. Customer Representative: Is there a dedicated customer representative involved in the project who can spare time for regular feedback and testing?
3. Testing: What is the nature and extent of testing required from the customer before we can move to final testing?
4. Release Expectations: What are the specific timeframes when the customer is expecting the releases?
5. Communication and Collaboration Tools: What tools are we currently using to facilitate communication and collaboration between our team and the customer...”

EC5: During every prompt testing phase, ChatGPT demonstrates awareness of stakeholder and managerial backing. It displays comprehension of the challenges and seeks further information about client interactions, testing procedures, release anticipations, and methods of communication and collaboration. The provided information is formatted as a memo. However, in the second round of prompt testing, no additional questions were made by ChatGPT.

In the second prompt simulated additional information about requested questions is given to ChatGPT.

EC6: Upon clarification, ChatGPT offers recommendations on enhancing customer relationships in the given scenario. These suggestions serve as valuable guidance for addressing the issue. Nevertheless, the quantity and nature of the suggestions fluctuate based on the test pattern iteration. Typically, the response encompasses actions pertinent to the goals.

In test round 3 prompt iteration is demonstrated. Additional requests to the previous prompts can be made and ChatGPT responds to the changes.

EC7: ChatGPT reacts to iterative prompts according to user directives and has the capability to comprehend further explanations.

PEC2: ChatGPT recognizes its function as a member of the steering group and delivers presentation as requested. It appears to comprehend the topic at hand and offers actionable suggestions. These suggestions typically seem applicable. However, there is significant variability in the actions proposed during each round.

4.3 Role Clarification

Clear role definition helps projects to overcome this pain point when there is uncertainty of the responsibilities and too much overlapping. Intention for this pattern is to provide guidance and workflow how project manager could utilize ChatGPT when defining the right roles for the project that help individuals to be accountable of specified tasks defined in the project.

Role clarification pattern

Contextual statements:

“As a [roleA] your task is to specify clear roles for the team. The project team has the following constraints: [c1, c2, c3...].

Ask all necessary questions to fully understand the project and be able to create necessary role specifications.

Explain your understanding and provide solution to me representing the [roleB]. Provide output in [format]. “

To further elaborate, the avoidance of role voidance and overlap is crucial as it ensures that all necessary project tasks are accounted for and that team members are not burdened with multiple, conflicting responsibilities. This pattern helps to streamline the project process by assigning specific roles to individuals based on their skills and expertise. By doing so, team members can focus on their assigned tasks and contribute to the project’s overall success.

Prompt Pattern Demonstration: *“As an engineering manager your task is to specify clear roles for the team. The project has the following constraints: software is created using C++ and publicly known open-source components. Only one person knows how to program in C++ but the rest 4 persons have experience only in Python and Javascript. Team needs to deliver C++ application in Linux environment in the next four months. Ask all necessary questions to fully understand the project and be able guide team to create necessary role specifications. Explain your understanding and provide solution to me representing the management. Provide output in Table format.”*

In the first round ChatGPT responses: “Sure, I’d first need to understand a few more details about the project to make the role specifications as accurate as possible. Here are the questions...”.

EC8: ChatGPT appears to grasp the subject and context and offers supplementary questions for further clarity. In every response cycle, it lists initial roles along with their respective descriptions.

For ChatGPT’s response the following second prompt is provided to clarify the project role needs: “Project needs to deliver C++ application in embedded device and transfer the data to backend. It should collect IoT data and move that to the backend for further processing. We use existing cloud-based backend but IoT device as Atmel based 32-bit processor and necessary hardware. We would like to utilize existing sw designers also in C++ development. We collaborate through GitHub using its features. Linux is ubuntu based. Testing is done fully manually as we don’t have suitable tools for testing C++ applications and the application is simple. We plan to make some error updates but otherwise maintenance is approx. Two times in the year.”

EC9. ChatGPT processes the supplementary prompt, seeking clarifications on its queries. Once it assimilates the provided information, it then curates a detailed presentation outlining the necessary roles for the project.

ChatGPT: “Based on the additional details, I suggest the following roles and responsibilities for your team members...”.

ChatGPT: "...The specifics of these roles might need to be adjusted based on the specifics of your team and your project, but this should give you a good starting point."

EC10: Based on the specific prompt test iteration, various role specifications are displayed. Moreover, in every cycle, ChatGPT underscores the potential need for modifications to the roles e.g.: "The specifics of these roles might need to be adjusted based on the specifics of your team and your project, but this should give you a good starting point."

In test round 3 prompt iteration is demonstrated. Additional requests to the previous prompts can be made and ChatGPT adapts responses to the additional information given.

EC11: ChatGPT responds to prompt iterations based on user instructions and can understand additional clarifications.

PEC3: ChatGPT appears to grasp the context of the prompt pattern and presents an initial role description, which includes the role's responsibilities and necessary skills based on the provided feedback. Moreover, it conveys that the roles may require adjustments in accordance with the actual requirements of the project.

4.4 Empirical Contributions

ChatGPT's ability to adapt and provide actionable insights is central to the ECs. EC1 focuses on ChatGPT's initial engagement, where it asks contextual questions and presents preliminary requirements in a table. EC2 offers a revised set of requirements based on additional user input. EC3 shows that ChatGPT can autonomously prioritize requirements, even without explicit instruction. EC4 and EC7 emphasize its adaptability to iterative prompts and its capability to understand further clarifications. EC5 and EC6 highlight ChatGPT's awareness of stakeholder and managerial support, offering actionable recommendations for enhancing customer relationships. EC8 through EC11 delve into role clarification, where ChatGPT not only asks additional questions for clarity but also provides a detailed outline of necessary roles, emphasizing that these may need adjustments based on specific project needs. Overall, the ECs demonstrate ChatGPT's versatility in adapting to user needs, understanding project complexities, and offering tailored recommendations.

ChatGPT's proficiency in understanding context and delivering tailored outputs is evident in the main results, **PECs**. PEC1 showcases ChatGPT's ability to seek clarifications and offer high-level requirements in a structured table format. PEC2 highlights its role as a steering group member, where it not only delivers the requested presentation but also provides actionable suggestions, albeit with some variability across iterations. PEC3 demonstrates ChatGPT's skill in role clarification, presenting initial role descriptions complete with responsibilities and required skills, while also acknowledging that these roles may need to be fine-tuned based on actual project requirements. Collectively, the PECs underscore ChatGPT's capabilities in offering structured, actionable insights while adapting to varying project needs and contexts.

5 Conclusions

In this initial study we have demonstrated how prompt engineering can be used to solve agile software management problems. We developed an APM pain point model and for each of the pain point, we have now crafted a prompt pattern that can be used to consult or even solve the problem related to the pain point. Three patterns were introduced in the paper. The future research looks forward to introducing more patterns.

References

1. Layton, M.C.: *Agile Project Management For Dummies*. Wiley, Hoboken (2012)
2. Manifesto for Agile Software Development. <http://agilemanifesto.org/>. Accessed 25 Oct 2023
3. Agile Practice Guide. <https://www.pmi.org/pmbok-guide-standards/practice-guides/agile>. Accessed 25 Oct 2023
4. Hoda, R., Murugesan, L.K.: Multi-level agile project management challenges: a self-organizing team perspective. *J. Syst. Softw.* **117**, 245–257 (2016). <https://doi.org/10.1016/j.jss.2016.02.049>
5. Zeitoun, A., Kerzner, H.: *Project Management Pain Points and a Path Forward*. Wiley, Hoboken (2021)
6. Introducing ChatGPT. <https://openai.com/blog/chatgpt>. Accessed 25 Oct 2023
7. What is generative AI? <https://research.ibm.com/blog/what-is-generative-AI>. Accessed 25 Oct 2023
8. Stephens, M., Vashishtha, H.: *AI Smart Kit: Agile Decision-Making on AI*. Information Age Publishing, Charlotte (2021)
9. Nieto-Rodriguez, A., Vargas, R.V.: *How AI Will Transform Project Management* (2023). <https://hbr.org/2023/02/how-ai-will-transform-project-management>
10. Poston, R., Patel, J.: Making sense of resistance to agile adoption in waterfall organizations: social intelligence and leadership. In: *AMCIS 2016 Proceedings* (2016)
11. Nuottila, J., Aaltonen, K., Kujala, J.: Challenges of adopting agile methods in a public organization. *Int. J. Inf. Syst. Project Manage.* **4**, 65–85 (2016). <https://doi.org/10.12821/ijispm040304>
12. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. *J. Syst. Softw.* **119**, 87–108 (2016). <https://doi.org/10.1016/j.jss.2016.06.013>
13. Gregory, P., Barroca, L., Sharp, H., Deshpande, A., Taylor, K.: The challenges that challenge: engaging with agile practitioners' concerns. *Inf. Softw. Technol.* **77**, 92–104 (2016). <https://doi.org/10.1016/j.infsof.2016.04.006>
14. Reunamäki, R., Fey, C.F.: Remote agile: problems, solutions, and pitfalls to avoid. *Bus. Horiz.* **66**, 505–516 (2023). <https://doi.org/10.1016/j.bushor.2022.10.003>
15. Paasivaara, M., Behm, B., Lassenius, C., Hallikainen, M.: Large-scale agile transformation at Ericsson: a case study. *Empir. Softw. Eng.* **23**, 2550–2596 (2018). <https://doi.org/10.1007/s10664-017-9555-8>
16. Sithambaram, J., Nasir, M.H.N.B.M., Ahmad, R.: Issues and challenges impacting the successful management of agile-hybrid projects: a grounded theory approach. *Int. J. Proj. Manag.* **39**, 474–495 (2021). <https://doi.org/10.1016/j.ijproman.2021.03.002>
17. Shameem, M., Kumar, R.R., Kumar, C., Chandra, B., Khan, A.A.: Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process. *J. Softw. Evol. Process.* **30**, e1979 (2018). <https://doi.org/10.1002/smr.1979>

18. Kamal, T., Zhang, Q., Akbar, M.A.: Toward successful agile requirements change management process in global software development: a client–vendor analysis. *IET Softw.* **14**, 265–274 (2020). <https://doi.org/10.1049/iet-sen.2019.0128>
19. Drivers of Project Agility | PMI. <https://www.pmi.org/learning/thought-leadership/pulse/drive-project-agility>. Accessed 25 Oct 2023
20. Guinan, P.J., Parise, S., Langowitz, N.: Creating an innovative digital project team: levers to enable digital transformation. *Bus. Horiz.* **62**, 717–727 (2019). <https://doi.org/10.1016/j.bushor.2019.07.005>
21. Barke, H., Prechelt, L.: Role clarity deficiencies can wreck agile teams. *PeerJ Comput. Sci.* **5**, e241 (2019). <https://doi.org/10.7717/peerj-cs.241>
22. Fu, M., Tantithamthavorn, C.: GPT2SP: a transformer-based agile story point estimation approach. *IEEE Trans. Software Eng.* **49**, 611–625 (2023). <https://doi.org/10.1109/TSE.2022.3158252>
23. Agile: The Human Factors as the Weakest Link in the Chain | SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-27896-4_6. Accessed 25 Oct 2023
24. vomBrocke, Jan, Hevner, Alan, Maedche, Alexander: Introduction to design science research. In: vomBrocke, J., Hevner, A., Maedche, A. (eds.) *Design Science Research. Cases*. PI, pp. 1–13. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-46781-4_1
25. Koppenhagen, N., Gaß, O., Müller, B.: *Design Science Research in Action - Anatomy of Success Critical Activities for Rigor and Relevance* (2012). <https://doi.org/10.5445/IR/1000055012>
26. White, J., et al.: A prompt pattern catalog to enhance prompt engineering with ChatGPT. <http://arxiv.org/abs/2302.11382> (2023). <https://doi.org/10.48550/arXiv.2302.11382>
27. Mastering ChatGPT: How to Craft Effective Prompts (Full Guide + Examples). <https://gptbot.io/master-chatgpt-prompting-techniques-guide/>. Accessed 25 Oct 2023
28. DSH, T.: *Mastering Generative AI and Prompt Engineering: A Practical Guide for Data Scientists* [eBook]. <https://datasciencehorizons.com/mastering-generative-ai-prompt-engineering-ebook/>. Accessed 25 Oct 2023
29. OpenAI: GPT-4 Technical Report. <http://arxiv.org/abs/2303.08774> (2023). <https://doi.org/10.48550/arXiv.2303.08774>
30. OpenAI Platform. <https://platform.openai.com/tokenizer>. Accessed 07 June 2023
31. Khan, S.: ChatGPT Memory: Adaptive Prompt Creation. <https://redis.com/blog/chatgpt-memory-project/>. Accessed 25 Oct 2023

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

