AGILE(OU1)

Name: Gazi Md Rakibul Hasan
Course Name: Software Engineering.

APRIL 4, 2024

Abstract

We have delved into two Agile methodologies: Scrum and Extreme Programming (XP). By utilising a Large Language Model (LLM), we performed a comparative analysis of these two Agile methodologies. The LLM-generated text analysis had several shortcomings, including insufficient details, inconsistency, etc. Thus, it is essential that human expertise critically observes LLM-generated text. Real-world business tends to be different from the textbook Agile approach due to the need to adjust to the needs and requirements. We discovered that blending and hybrid Agile methodologies are more efficient in software development.

Introduction

This report aims to discuss the two agile development approaches, Scrum and XP, through an extensive literature review. Afterwards, we will use an LLM to compare these Agile methodologies. Our goal is to identify and discuss the shortcomings of these generated texts. Then, we will reflect on the practical implications of agile methodology and why organisations may diverge from the "textbook" approach to accommodate real-world constraints.

Background

Traditionally, the software development process was dependent on the Waterfall methodology. That model consists of several sequential stages, making it a plan-driven process. The problem emerges within the waterfall methodology when new information is included while developing these phases or any information needs modification. Then, the whole process must be backtracked. It is costly and time-consuming. (Sommerville, 2016).

To address this problem, industry experts developed principles and values that allow software development to be more flexible to changes. As a result, they have manifested The Manifesto of the Agile Alliance. It includes the following statements,

Individual and interactions over process and tools: People are the most essential elements for a project's success. A pack of average programmers who communicate and interact well are likelier to succeed. The right tools are essential, but there are cases in which the tools are more emphasised.

Working software over comprehensive documentation: Software development is perilous without documentation. However, more excessive documentation is harmful. Therefore, the documentation must be short and salient.

Customer collaboration over contract negotiations: Successful projects involve frequent customer involvement and feedback. It is not ideal to depend solely on a contract or statement of work.

Responding to changes over the following plan: The software development plans must be flexible and ready to adapt to changes since they are detrimental to the success of a software project. The requirements change for many reasons, such as the business environment can change, and the customers are likely to alter the requirements once they see the system start functioning. (Martin, 2003).

Scrum

The Scrum Agile methodology offers a framework for structuring Agile projects and facilitates external visibility of the project's progress. Scrum does not include any obligatory usage of any specific practices. Consequently, Scrum can be easily integrated into an organisation's existing procedures. (Sommerville, 2016).

The Scrum framework comprises several roles: product owner, Scrum master, and development team. The product owner is responsible for deciding what will be developed and the sequence in which it will be done. The ScrumMaster leads the team and conforms to its process development. The development team determines how to deliver what the product owner has asked to develop. (Rubin, 2013).

The product owner has a vision, which can be broken down into a set of features that are collected into a prioritised list called the product backlog. These items could be user stories or instructions for the refactoring of codes. High-prioritise items will be placed at the top of the product backlog. The scrum process is referred to as the sprint cycle. The starting point for the scrum sprint point is the product backlog's item. Each iteration process of the sprint cycle produces a product increment for the customer. Each sprint cycle has a fixed amount of time, which could last from 2 to 4 weeks. The product owner's responsibility is to ensure that details in the product backlog's items are appropriate for the work to be done and prioritise the items that will be developed in the cycle. However, an item will be returned to the backlog if the item cannot be developed within the allocated time frame. All team members get involved while selecting the highest priority item they believe they can complete and its estimated time for development. To estimate the time, they utilise the velocity that was attained in the previous sprint. It leads to the creation of the sprint backlog. Scrums(daily meetings) are held daily, and all team members share their progress and information, discuss the problem, and re-plan with other team members if any problem arises. The whiteboard is a shared resource for the whole team, and it is used to post notes about the sprint backlog. It includes the work to be done, unavailability of the stuff and so on. Anyone in the team can move or change items from the whiteboard. With the help of the whiteboard, anyone can see the team's current progress. The ScrumMaster reports the progress, longer-term planning, and budget to the senior management. Furthermore, the ScrumMaster is responsible for the project administration. (Sommerville, 2016).

Extreme Programming

Kent Beck introduced Extreme Programming (XP) in 1998 to improve software development and software development practices. XP emphasises agility and flexibility, which are closely related to the principles mentioned in the Agile Manifesto.

User stories: User stories describe scenarios where users interact with a system, detailing their actions to achieve a specific goal. Story cards will be developed based on the user stories. The story cards can be broken down into tasks, and the developer implements those tasks. However, the customer will first prioritise which task will be developed. When the requirement changes, the unimplemented story cards will be discarded or modified, creating new stories.

Incremental Development: The system will be frequently updated with minor updates. The requirements are presented as user stories, and the team will decide which functionality to include based on the user stories.

Customer Involvement: The customer will participate in the development and collaborate to define the system's acceptance tests.

Pair Programming: Programmers work in pairs. Pairs are created dynamically so all team members can work together. Pair programming supports the collective ownership and responsibility for the system. Since pair programming allows the code to be reviewed by at least two people, it leads to less error in the code and can easily refactored. However, many companies are suspicious of the efficiency of pair programming.

Test-first development: In XP, tests are automated and created before the development process is implemented. The customer is involved during the test's creation and validation. The test gradually develops from the user stories.

Refactoring: XP practice requires that the code be simple and maintainable. Therefore, the code is constantly refactored when there is room for improvement. Refactoring improves the software's readability and maintainability. (Sommerville, 2016)

Discussion of comparison

To identify the shortcomings of LLM-generated text, we inserted the following commands: "Why Scrum is better than XP" and "Why XP is better than Scrum" in Chat-GPT. (OpenAI, 2024). The output of these commands is illustrated in Figure 1 and Figure 2, respectively.

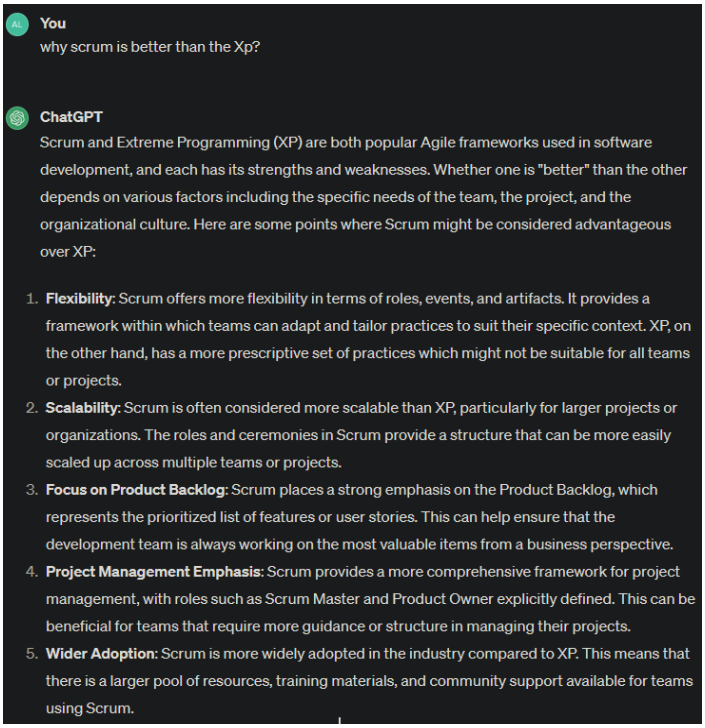

Figure 1: The result of the command "Why Scrum is better than XP?" in Chat-GPT.

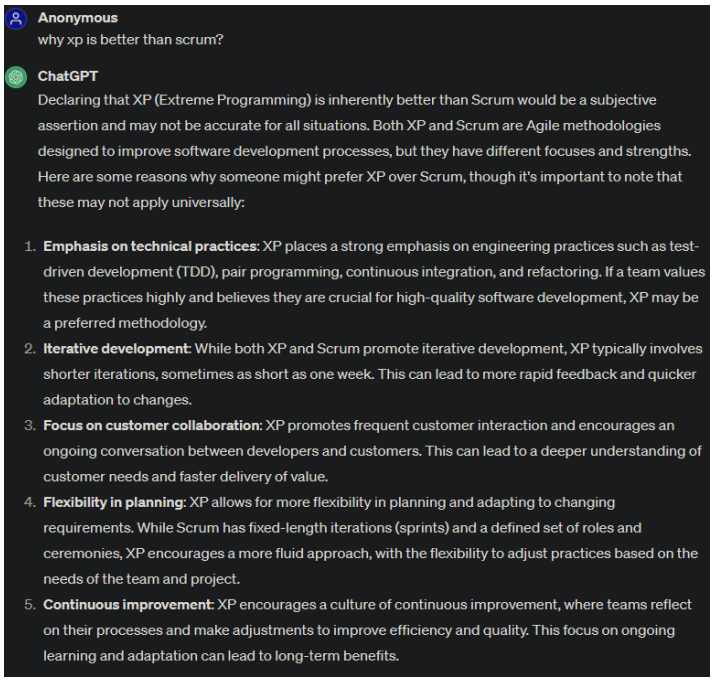

Figure 2: The result of the command "Why XP is better than Scrum?" in Chat-GPT.

Depth and Details: As mentioned in Figure 1 in Scrum methodology, the items will be placed in a product backlog. As we know from our literature study, the highest prioritised item will be placed at the top of the product backlog, and it did not mention explicitly how the product backlog is formatted in Figure 1. Meanwhile,

Figure 2 shows that XP is better than Scrum in terms of technical practices. Thus, if a company prefers high-quality products, it should use XP. However, that statement lack details on these terms, such as pair programming, test-driven development, and so on.

Practical challenges: In Figure 1, Scrum is regarded as being more flexible in terms of roles, events, and artefacts. However, a new member lacking knowledge of Scrum may find integrating with the Scrum team hard. (Patidar, 2024). In contrast, XP contains practices such as pair programming, facilitating more mentorship for new team members.

Comprehensive coverage of Agile principles: Scrum and XP are regarded as agile methodologies, so both should align with the Agile manifesto principles. As we can see, both Figure 1 and Figure 2 mention that they follow the iterative development process, which is a core aspect of Agile software development. Nevertheless, it is noticeable that the absence of how these methodologies integrate other Agile principles into their practices and which methodology would be more beneficial from an Agile principle's perspective, especially when adapting to changes in the requirements.

Integration with other methodologies: Organisations should adopt blended, or hybrid approaches rather than strictly adhering to a single methodology. This approach enables organisations to adapt in a way which is more beneficial for them. (Ozkan, Erdogan, Bal & Gök, 2022). It implies the importance of the integration of various methodologies. As we can see from both Figure 1 and Figure 2, none of the comparisons mentioned which methodologies best fit to Scrum and XP.

Inconsistency: Figure 2 states that if companies prefer high-quality software development, XP should be the preferred methodology. Conversely, as stated in Figure 1, Scrum is widely adopted in the industry compared to XP. It creates a paradox since it is evident that all organisations aim for high-quality software development. Now, why Scrum is widely adopted in the industry when XP is regarded as a methodology for high-quality software development arises. It implies the inconsistency of the LLM-generated text.

Critical Reflection

Large organisations have been running for years in the traditional approaches; fully transitioning to the Agile approach is problematic. The study found that the team had difficulty collaborating with team members who had difficulty being committed to the project when the complete transition was applied suddenly. However, blending Waterfall and Agile methodology is the most effective approach to ease the transition and cost expenses. (Wankhede, 2016). Each Agile methodology has its limitations; for instance, in Scrum, story points can be misinterpreted and misused. The story point intends to help the team estimate the required time to complete different tasks. The XP encourages collaboration among team members and customers. It requires high discipline, communication and trust among team members, which can be challenging. It is proposed that these methodologies be blended to mitigate the weakness of an Agile methodology pose. For example, a team can use Scrum's time boxing and XP's technical practices simultaneously. (Incedayi, 2023). The software industry utilises hybrid methodology in software development to enhance competitiveness and improve the quality of the software product. The combination can consist of methodology, quality, and managerial standards. (Sarpiri, M.N. , T.J. Gandomani. , 2020).

Conclusion

We can conclude that comparing the Scrum and XP with the LLM-generated text has significant flaws in depth and details, practical challenges, comprehensive coverage of Agile principles, interaction with other methodologies and inconsistency. Therefore, one should be cautious about using LLM tools such as Chat-GPT. It seems essential to have human expertise intervene in extracting and interpreting information from the LLM generative tools. The Agile methodologies we discussed in that report are Scrum and XP, which have their weakness. Therefore, software development is most likely different from the textbook approach in the real world due to its adaptive nature and needs. Furthermore, we found that hybrid and blending of the Agile methodologies are more efficient than a particular Agile methodology.

References

Sommerville, I.,(2016). Software Engineering. 10th ed. Boston: Pearson, ISBN 978129209613.

Martin, R.C., (2003). Agile Software Development, Principles, Patterns, and Practices. 1st ed. Upper Saddle River: Pearson Education. ISBN 13: 978-1-292-02594-0.

Rubin, K.S., (2013). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Pearson Education, Inc. ISBN 978-0-13-704329-3.

OpenAI. (2023). ChatGPT. Available at: https://chat.openai.com/ [Accessed 03-04-2024].

Patidar, K., (2024). Common Challenges in Scrum Implementation and How to Overcome Them. LinkedIn. Available at: https://www.linkedin.com/pulse/common-challenges-scrum-implementation-how-overcome-them-patidar-krmof/ [Accessed 03-04-2024].

Ozkan, N., Erdogan, T.G., Bal, S. and Gök, M.S. .(2022). Scrum, Kanban or a Mix of Both? A Systematic Literature Review. Information Technologies Research and Development Centre, Sakarya University, Hacettepe University, Gebze Technical University.

Wankhede, R. (2016). Hybrid Agile Approach: Efficiently Blending Traditional and Agile Methodologies. Harrisburg University of Science and Technology.

Incedayi,k. (2023), Kanban, XP, Scrum, or a mixed platter? Medium. Available at: https://medium.com/@cevatkerim/kanban-xp-scrum-or-a-mixed-platter-379c9e99d5d6 [Accessed 03-04-2024].

Sarpiri, M.N. , T.J. Gandomani. (2020), A Case study of using the hybrid model of Scrum and Six Sigma in software development. Islamic Azad University, Shahrekord University.