University of Sindh Journal of Information and Communication Technology (USJICT)
Volume 7, Issue 2

ISSN-E: 2523-1235, ISSN-P: 2521-5582
Website: http://sujo.usindh.edu.pk/index.php/USJICT/

A Study on Agile Retrospective Practices in the Software Industry of Pakistan: An Examination of Real-World Applications

Ali Raza, Jaweria Kanwal, Maryam Imtiaz Malik, Fatima Gillani

Department of Software Engineering, National University of Modern Languages, Islamabad, Pakistan
Ali.raza4873@gmail.com, jaweria.kanwal@numl.edu.pk, Maryam.imtiaz@numl.edu.pk, Fatima.gillani@numl.edu.pk

Abstract: In every industry, manual processes are transitioning to software technology. With rapid technological advancements, software development organizations are compelled to continually update and enhance their development processes. Agile practices in software programming emphasize client satisfaction and adaptability to change. Retrospective practices play a crucial role in process improvement within Agile software development, fostering greater collaboration among team members. Through retrospective practices, challenges and issues within the development process are identified, leading to the implementation of appropriate solutions. Team velocity is managed based on the progress of previous sprints. While retrospective practices may initially pose challenges for new teams, typically taking 3-5 meetings to fully adopt, they are instrumental in fostering a mindset conducive to productive retrospective meetings. A case study was conducted to assess the adoption of retrospective practices within the Pakistani software industry. The findings reveal that while retrospective practices are indeed utilized for process improvement, they yield both positive and negative impacts on software quality and team enhancement. Pakistani software teams implement a variety of retrospective practices tailored to meet specific software quality requirements, with overall satisfaction among team members regarding the practices employed within their respective teams.

Keywords: Retrospective Practices; Process Improvement; Collaborative Games; Software Quality; Agile; Scrum;

I. INTRODUCTION

Agile software development has gained widespread recognition as a mainstream practice in the global software engineering community, having been a prominent fixture in the IT industry for over three decades. Among the array of agile methodologies, Scrum stands out as the most widely adopted [1]. Agile methodologies are characterized by their flexibility, iterative nature, and emphasis on swift adaptation to changes, distinguishing them from traditional software development approaches [3,4]. Due to their demonstrated efficiency benefits, agile practices have become commonplace in software development [5,6,8].

Agile methodologies refer to a set of evolutionary and iterative methods, which are based on opportunistic and iterative enhancement processes of development. The acknowledgment of Agile strategies for the purpose of product improvement was decidedly impacted in the year 2001 [9,11]. Agile philosophy presents the coordinated system, express the qualities and standards of all agile practices [10,12]. Agile practices are used for the process improvement in software industry but it is difficult to accept agile practices in development environment completely [2, 3, 7].

Moreover, integrating game elements into non-gaming contexts has emerged as a focal point for advancing product enhancement methodologies [13, 14]. Studies have validated that gamifying tasks within software development can mitigate certain challenges; however, it may also present obstacles such as discordance among team members, limited collaboration, and a lack of motivation [15, 16, 17].

To address these challenges, agile mentors have suggested conducting review sittings in the form of games, aiming to disrupt the daily paractice and enhance engagement during the session [18]. However, limited research has been conducted thus far to consider the impact of review games on the effectiveness of such meetings [19].

In agile process improvement practices, specifically the Scrum, have become norm in proficient software improvement process [20]. The Retrospective meetings is a well-known process improvement approach proposed in Scrum. It is a formal meeting held toward the completion of a Sprint to assess the progress of the team, to identify the gains and declines and possible improvements. That's what kniberg claims "the retrospective is the main most-significant thing in Scrum" [21]. In Scrum teams, 80% team members made plans for using extra agile methods for the purpose of team collaboration and sprint planning for scaling Scrum like large scale agile frameworks for process improvement [22].

In the current era of software development demand of software quality and efficiency increase day by day. Software developing teams need to improve process to ensure the quality and efficiency in software so that they can satisfy their customers [4,59,60]. In software development, Scrum is the mostly used methodology because it focuses on the completion of software within time and budget [58]. But Scrum does not explicitly focus on the quality assurance activities during software development as traditional methodologies does. There are some agile practices that assess the positives and negatives of the development team at the end of the sprint. Retrospective practices are one of them. The fundamental goal of this research is to acquire a comprehension of interaction improvement through retrospective practices and approaches. We focus on to identify the retrospective practices and approaches that are used in Pakistan software industries and ways to proceed these retrospective practices with in the software organizations in Pakistan. For this purpose, we conduct a case study in a software company named as “10 Pearls” with the project manager and the development team. The team has been using these practices for 8 years in software industry.

The remainder of the document is structured into various distinct sections. The next section is background about the Agile, Scrum and retrospective practices and collaborative games. In section III related work is presented and in section IV represents research methodology. In section V results are presented and we discussed the results in Section VI. In section VII, we concluded the paper.

II. RELATED WORK

The necessity for continuous process improvement is one of the guiding principles of all agile methodologies: " The team considers ways to improve effectiveness on a regular basis and modifies its behavior in response." (Highsmith and Fowler 2001) [9,11]. An agile retrospective meeting is often conducted according to a defined framework that includes setting the stage, collecting data, producing insight, choosing what to do, and concluding the meeting [2]. During retrospective sessions, team members engage in sharing their evaluations and feedback regarding the previous sprint's accomplishments. Agile teams convene to deliberate on priority issues, devise resolutions, and strategize improvements. These sessions are regarded as reflective opportunities enabling team members to evaluate their performance and enhance self-awareness concerning work-related challenges and their problem-solving strategies. This heightened awareness fosters avenues for professional advancement and skill development. Researchers often classify these reflective practices into different levels, commonly including reporting and responding, relating and reasoning, and reconstructing. Some studies discuss more about general retrospective issues [49], while some associated the issues to explicit retrospective games [50]. The proposed planning was assessed by industry and students in a subsequent report [51].

The use of cooperative games in programming improvement is getting increasingly more consideration. A significant foundation for this examination region was the games presented by [44] as market and item research strategies and later embraced by [46] and [47] to improve necessities designing cycles. [48] Call attention to, that coordinating programming process improvement exercises in deft advancement conditions is not quite the same as in additional conventional ones.

The gathered results suggest that retrospective games discuss topics according to the plan. The diagnosing phase of our ongoing study while setting up the meeting convention made use of the retroactive issues identified in the earlier work. Additionally, results-focused analysis supports that game-based reviews primarily resolve common retrospective problems [4]. Utilizing retrospective sessions in Agile programming enhancement can facilitate the iterative improvement of various aspects that may impact the final product's quality or the work environment. Incorporating interactive exercises or activities, such as games, during these sessions can foster a positive mindset within the development team. Derby and Larsen [34] gave the plan five sorts of games that could be utilized successively in a similar review meeting: set the stage, accumulate information, produce bits of knowledge, choose

what to do, and close the review. Gonçalves and Linders [42] and Caroli and Caetano [53] depicted separately 13 and 44 games that can be utilized to work with reviews. In the writing different 85 [52] games or practices that utilization during the agile retrospective meetings. Some of them that are major utilized as retrospective game/practice referred to by creators are given in the Table 1 momentarily shows the review games that we utilized in our earlier examinations.

III. BACKGROUND

In this section, we present key concepts pertinent to our study's context: agile methodologies, Scrum framework, agile retrospectives, as well as games and retrospective activities. The operational framework of Scrum is illustrated in Figure 1.

A. Agile

Agile is a continuous way to deal with project management and software development that assists the team to facilitate their clients faster. Rather than putting in all on the line "big bang" sendoff, an agile team convey work in small, but consumable, increases. Requirements, plans, and results are assessed constantly so teams have a characteristic system for answering change rapidly [24].

The agile philosophy focuses that during the development process customer and developers communicate with each other with a working software rather than a document and welcome changes in the development throughout the development process [24].

B. Scrum

Scrum is a lightweight agile management framework that aids in the management of software development operations through incremental and iterative processes. Scrum is used in businesses to adapt to changing client needs. Scrum emphasizes "What can be done" rather than "Why it can't be done." Client input is collected on a regular basis, which aids the company in controlling and mitigating risk at an early stage. Sprints are used to coordinate the development process. A sprint meeting is held at the start of each sprint process, during which the sprint time is determined and the development team picks high-priority features from the product backlog. Scrum is a methodology that assists personnel, groups and organizations to provide versatile solutions for complex problems [26].

1) Scrum Cermonies: Recommended events are utilized in Scrum to make routineness and to limit the requirement for gatherings not characterized in Scrum. These Scrum ceremonies are given below [26]:

- Sprint
- Sprint Outlining
- Daily Scrum
- Sprint Analysis
- Sprint Retrospective

a) Srpint: A Scrum Sprint encompasses a timeframe of one month or less, within which a fully completed, usable, and potentially releasable increment is developed.

b) Sprint Outling: The sprint starts with planning by arranging the work for a sprint to be performed. This subsequent arrangement is made by the cooperative work of the whole Scrum Team.

*c) Daily Scrum:*The motivation behind the Daily Scrum is to investigate progress toward the Sprint Goal and adjust the Sprint Backlog as required, changing the impending arranged work.

d) Sprint Analysis: The purpose of the Sprint Review is to evaluate the sprint's outcome and determine any changes that should be made going forward. The Scrum Team talks about the progress made toward the product goal and shares the results of their work with important partners.

e) Sprint Retropsective: The Sprint Retrospective is meant to help develop strategies for increasing both quality and sufficiency. The Sprint Retrospective finishes up the Sprint. It is timeboxed to a limit of three hours for a one-month Sprint. For more limited Sprints, the occasion is generally more limited.

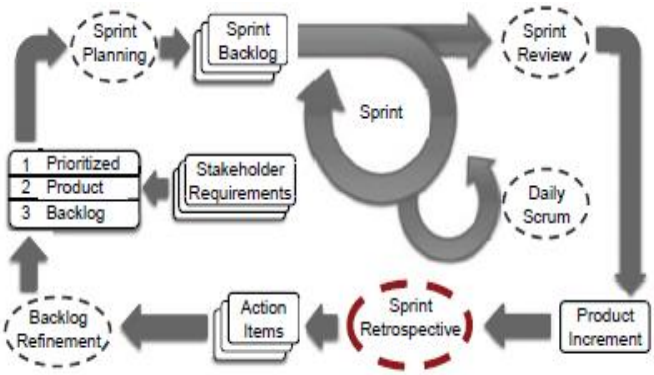

Figure 1. Flow of the Scrum process, based on [23].

TABLE I. RELATED RESEARCH ON RETROSPECTIVE PRACTICES

Game	Game Components	Description	Ref
Sailboat	A depiction featuring a sailboat, rocky terrain, gusts of wind, and an island.	It additionally assists the group with recognizing obstacles, potential dangers, and things that cause them to convey extreme software	[4,42]
Mad/Sad/Glad	Three columns	IT helps discharge a weighty touching steam and associate colleagues' feelings to occasions that occurred in the Sprint	[4,34]
Mood++	five columns	Enhance Mad/Sad/Glad by introducing two new types	[4,54]
Starfish	an image of a starfish with five arms	In opposite to Mad/Sad/Glad, they resemble the group to thought mostly from an impartial point of view, rather than an emotional point of view.	[4,42]
5Ls	five columns	In opposite to Mad/Sad/Glad, they resemble the team to thought mostly from	[4,54,55]

		an impartial point of view, rather than an emotional point of view.	
360 Degrees	Circle Shape and a paper for each member	The game gives an amazing chance to recognize colleagues' diligent effort, commitment, and help. It is particularly valuable after an intense cycle to raise camaraderie	[4,53]

C. Agile Retrospectives

One of the important standards of all agile practices is the requirement for ceaseless cycle improvement: " During regular intervals, the group thinks about how to become more viable, then adjusts and modifies its behavior accordingly" [11].

In Scrum, a practice is followed by the Scrum team where all the members of the team meet up to review, fix, and further develop their cycle. This practice is called a review meeting. This is known as the sprint review in the Scrum framework [27]. The social-mental component entails team members thinking on the goals, practices, and procedures of the group as well as the larger organizational environment and surrounds, and then adjusting as necessary (West, 1996). The Scrum team holds a meeting known as the Retrospective meeting following the conclusion of the Sprint and Sprint Review. Everyone in attendance is expected to fill out the survey and take part in the conversation. During this meeting, three key requests/questions should be discussed [28]:

- What went well, which makes the Retrospective start out well, and which allows us to appreciate all the excellent things that have happened?
- What didn't work, and how could we improve it?
- What we've learned motivates us to consider what we've discovered about how we work?

In like manner, the review is a chance for the group to think back over a cycle and perceive victories and deficiencies; to connect the relevant insight to individuals, the improvement cycle, designing practices, and devices; and to make an arrangement for enhancements to be established during the next iteration [29,30,31]. To be sure, the retrospective is one of the most often referenced agile practices with regards to programming process improvement [32,33]. The result of a review is "things to do" [34], which characterize distinguished issues as well as could be expected restorative activities to determine them [35].

D. Games

Games have been a piece of human progress for millennia [15] and have helped in making day-to-day existence and reality seriously fascinating and drawing in [36]. As per [37], playing games is one of the actions that assist members to acquire a perspective which is portrayed by the complete entry into the thing they are doing, and a consequently changed sensation of time. This psychological state where an individual playing out a movement is totally submerged in a sensation of stimulated concentrate, full commitment, and pleasure during the time spent the action, is called stream [37]. Later on, [38] extended stream hypothesis and contended that a team can enter a condition of stream to expand their modernization. The renowned book by [39] titled “Theory of games and economic behaviour” is the foundation whereupon cutting edge game hypothesis is based. As indicated by game hypothesis, games fall into three general categories [40,17].

1) Categories of games: There are three categories of gams given below:

a) Cooperative Games: Where people structure techniques that straightforwardly go against to other people.

b) Competitive Games: Where players have interest that are neither totally gone against nor totally incidental; subsequently, potential open doors exist for coopration however not really for equivalent award.

c) Collaborative Games: Where all players cooperate collectively, sharing a shared objective, and punishments.

[41] instituted the expression "serious games" to state games which "are not planned to be played principally for entertainment," but rather are intended for teaching or taking care of an issue.

E. Retrospective Games / Practices

Review games are a type of cooperative games utilized for agile retrospective. They allude to a few organized social exercises motivated by game play and intended to assist the members with sharing their insight and encounters on a given subject, investigate this information in manners that may not happen during typical cooperation, distinguish stowed away suppositions, foster a joint comprehension of an issue, and produce creative thoughts regarding taking care of the issue [35, 42, 43]. Exercises like moving sticky notes, writing on white sheets, gathering things, or drawing pictures urge fewer vocal individuals to add to the conversation, assist with beating restraints, and cultivate innovative reasoning [44, 45] by animating option mental cycles. They have additionally been professed to assist agile teams with obtain improved results from retrospective meetings [18].

IV. RESEARCH METHODOLOGY

In this section, we present our research methodology and research questions to understand the retrospective practices followed by Pakistani software industry. Complete research process follows shown in the Fig.2.

A. Objectives

Main objective of this research is to identify retrospective practices and approaches used by the teams of Pakistan software industroy. As retrospective practices are followed to identify the points of improvement, this study will eventaully help in improvement of software and sftware development process.

B. Research Questions

To fulfill our research objectives, we defined the following research questions:

RQ1: What is goal of using retrospective practices and approaches?

RQ2: Which practices / approaches are following as retrospective practices?

RQ3: How retrospective practices / approaches impact on software quality and software development team?

RQ4: What are advantages and disadvantages of retrospective practices?
RQ5: How satisfied are the team members with the use of Retrospective practices and which practices are most likely by them?

C. Cases Study

As our research is a qualitative study, we follow the case study method and conduct formal interview with a development team. An interview and survey-based study are conducted with the software industry named "10 Pearls". The company thoroughly follows retrospective practices in the Agile development and improvement process.

D. Data Collection and Data Analysis

This interview is a semi-structured interview and open-ended questioning throughout the interview. All the answer to our asked questions is noted and compared with the data in the literature about retrospective practices. We have also conducted a survey with the team members of the software industry to collect data regarding Retrospective practices. In the next section, all the results for each research question are discussed.

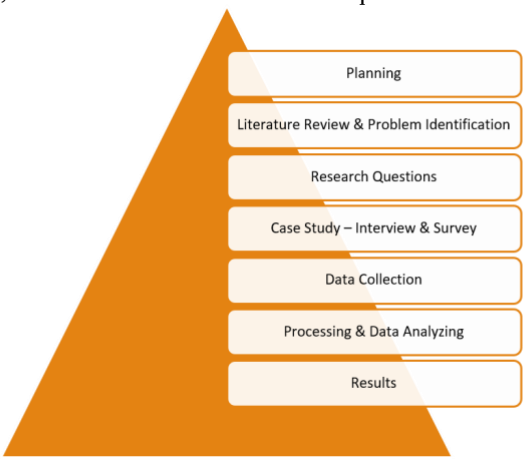

Figure 2. Research Methodology.

V. RESULTS

This section represents the results of our case study conducted with project manager and development team of "10 Pearl" software company. In the following, we present detail results and analysis of the research questions:

RQ1: What is goal of using retrospective practices and approaches?
There are different ways to improve the process of software development. Retrospective practices are one of them in Scrum development. In this section we discuss some practices that are followed as retrospective practices for the process improvement by the Scrum team in our case study. The main purpose of these retrospective practices is enhancing the communication and collaboration of the Scrum team members. The practices that are followed in the retrospective meeting are given below:

- What challenges and problems were faced during the sprint?
- How to overcome these problems and challenges?
- How to improve the development process?
- Any need of changing in process or team
- Highlight the mistakes
- Engage whole the team in good environment according to the goal of the projects

RQ2: Which practices / approaches are followed as retrospective practices in Pakistan software industry?
There is a variety of retrospective practices followed in Scrum. According to our case study analysis, retrospective practices and approaches vary from project to project and team to team and sometimes these are planned according to the budget and timeline of the project. The most common practices used by the Scrum team in Pakistani software industry according to our case study are GoRetro and 4L's. Cardwriting and starfish are also used depending on the above mentioned factors. Detail of these practices and their benefits are given in the following:

A. GoRetro Retrospective Practice

GoRetro is an online tool for managing retrospective meetings. It facilitates the entire retro process, make it simple, collaborative and data driven [57]. All the process remained on board for every retrospective meeting. GoRetro is mostly used by the Scrum teams according to our case study analysis because of its simplicity, accessibility and better communication medium. Team leads/managers can assess the progress of team, individual members of the team. It also provide features that help in analyzing team sentiments that how team members feels about the specific aspects of the product. Working pattern of GoRetro tool is as follows:

- All team members are given a piece of card like sticky notes to comment and turn it
- No limit for cards and columns to provide comments
- Voting facility for team member to give their opinion on problems
- Team member easily manage the cards to column and deal according to the need

B. 4L's Retrospective Practice

1) The 4L's method effectively manages both the favorable and unfavorable aspects of the Sprint while also fostering ongoing enhancement. Prior to commencing the activity, the facilitator partitions a board into four columns or displays four labeled posters [59]:

2) Liked: What did the development team like in the sprint about the processes that followed in the development?

3) Learned: What did the team learn new points from the sprint.

4) Lacked: What did the activities and tasks performed by the team is better in the sprint?

5) Longed For: What did the activities and tasks that team want to perform but not present during the sprint.

4 L's retrospective practice is also frequently used by the Scrum teams because in this practice, progress of all team members can be traced individually. Team leads/project manager can assess the team improvement such as whether team velocity is improving or not, which team member is working better on a project or there is need to change any team member from a project progress.

C. Starfish Retrospective Practice

It is a data-gathering activity to foster the thinking around practices and the value the team gets from them. It helps team members understand each other's perceived value on such practices [56]. Following are the set of actions in the Starfish retrospective practice.

1) Start: The activities and tasks that are not part of the development process should start and make part of the developemnt process.

2) Stop: The activities and tasks that are not benefical for the development process should remove from the tasks list and stop working on such activities

3) Continue: The processes and activites are doing well in the development process no need of changing in the working process.

4) Less: Effrots required to perform activites and task much smaller then the benefits. you see some value, but would rather cut down a little bit.

5) More: Activites and tasks that perform by a team in development process focused more and perform more than usual so more results and benfits comes from such activites and tasks.

D. Card Writing Retrospective Practice

The purpose of the card writing retrospective practice is that every team member should communicate in the meeting and the work performance of each member should be placed in front of all team members. The following steps are followed for the action of this retrospective practice:

- A piece of card is given to each team member
- Everyone has to write for every team member either positive or negative about his/her working performance
- Team leads or project manager collect all the cards form team members
- Team lead will read it loudly in front of everyone

RQ3: How retrospective practices / approaches impact on software quality and software development team?

As we discussed in RQ2, retrospective practices vary from team to team and project to project according to the allocated budget and time period, there is need to understand that how these practices impact the software quality, software process and the development team. Our analysis reveals that, overall, retrospective practices have positive impact in all the aspects. Following major points that are relevant to software quality, process improvement and team are given in the list below:

- Every team member communicates and discus issues and challenges regarding project he/she is facing.
- Problems and challenges are resolved in the retrospective meeting.
- Risks are identified and mitigation plan are prepared.
- A sprint postmortem is performed and in the retrospective meeting progress of the project is monitored.
- Manage velocity and retrospective meeting plan according to the project quality demands.

On the other hand, impact of retrospective practices on the development team at the start of retrospective practices with a new team is challenging and team members have no mindset about the retrospective practices. Initially 3-5 retrospective meetings have no positive impact on the new teams and gradually team make mindset with retrospective practices. Some positive factor for retrospective practices are given below:

- Appreciation of team increases
- Everyone can participate in all meetings and give his point of view
- Team performance improves and no one face biasness
- Discussion on each phase of development in front of whole team

RQ4: What are advantages and disadvantages of retrospective practices?

Retrospective practices play an important role in software improvement but as we discussed earlier, sometimes it becomes difficult to adapt such practices by the teams and organizations. We discussed about the advantages and disadvantages of retrospective approaches practiced by the project manager and team members of "10 Pearls" to know the perception of Pakistani industry. Advantages and disadvantages of retrospective practices are listed below:

E. Advantages

- All team members have clarity with the objectives and goals of the meeting
- Challenges and problems are identified
- Appropriate solutions are identified for the challenges and problems
- Collaboration among team members increased and relationships among them are strengthen
- Risks are identified and mitigated
- Project management process is improved
- Development process is improved

F. Disadvantages

- Retrospective practices are time consuming
- Proper planning and execution is required
- Difficult to make mindset of team towards retrospective practices
- Sometime teams have been built but could not deliver to improve processes
- Title of games are confusing for non-native English reader

RQ5: How satisfied are the team members with the use of Retrospective practices and which practices are most likely by them?

This research study is conducted with the Pakistan software industry for process improvement in software development. Retrospective practices are used for process improvement in the software industry. The statistics for the satisfaction of using retrospective practices are shown in Fig. 3. According to the results, all team members (100%) are satisfied with the retrospective practices. The results indicate that these practices are compatible for the process improvement of the software industry in Pakistan.

Fig 4 shows the responses of team members on the likeness of retrospective practice already followed in their industry. The result shows that 70% of members give preference to the GoRetro tool, 20% to 4 L's and 10% members give preference to card writing. These results depict the preference of team members working in '10 Pearls' software industry of Pakistan.

VI. DISCUSSION

Software development increase day by day and being a part of every industry. Agile software process development is followed in the software industry. Customer demand a high quality and efficiency with every project. In agile practices scrum frame work followed. The purpose of this study to recognize the practices and approaches that are used for the process improvement of the software product in agile methodology in

Pakistani software organization. As well as how to use retrospective practices and approaches followed in Pakistan.

Figure 3. Satisfaction with retrospective practices

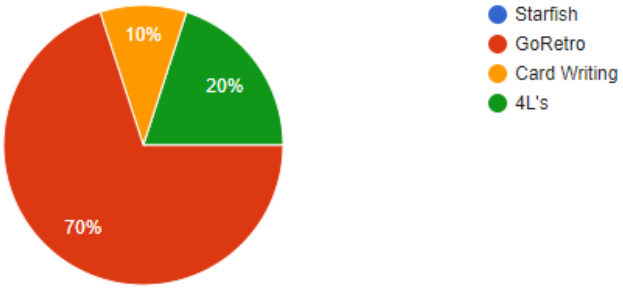

Figure 4. Best Retrospective practices according to team

Retrospective meetings are compulsory part of the scrum framework and these practices are following in scrum framework. Retrospective meetings are planned according to the time frame and budget of the software product. Retrospective practices and approaches vary from project to project or team to team. In the retrospective meeting complete postmortem of the sprint is performed and all problem and challenges are identified. Due to the Retrospective practices the communication and collaboration among the team increases. Each team member participates in the meeting and discusses the problems. The appropriate solutions for the problem are also identified in the meeting.

VII. Conclusion

Agile software development is a process that mainly focus on the customer satisfaction. This software process model warmly welcomed the changes in the project even in the last phase of the development. Scrum is one of the agile practices and retrospective meetings are the mandatory part of the scrum. Retrospective meeting is used for the process improvement of the program development and to better the communication between the development team. So, case study – interview and survey are conducted for this research with the project manager of a software company name “10 Pearls” in Pakistan. The RQ1 in the study concludes the goal of the retrospective practices and the factors noted in the retrospective meeting. These factors include the discussion of challenges and problems occur during the sprint and the suitable solutions for these problems. The engagement of each team member is also observed to facilitate the participation of each member equally. The results for RQ2 indicate that retrospective practices including Starfish, 4L’s, card writing and the GoRetro tool are used in the Pakistani software industry. Planning and use of retrospective practices vary from project to project and from time to time with the demand for quality. RQ3 explores the impact of these practices on software quality and development team. Retrospective practices have both positive and negative impact on the process improvement and team. The whole team communicates on the identified problems and finds a suitable solution for these issues and problems. Risks are identified and mitigated. Retrospective practices are difficult for new teams and has negative impact on the team average after 3-5 meeting team mindset set on the retrospective meeting. RQ4 investigates the benefits and drawbacks of the retrospective practice. These practices improve the process quality. Collaboration between the team also increases. Some drawbacks are difficult to make mindset of the team for retrospective practice and these practices are time consuming. The result for the RQ5 shows that all team members are satisfied with the retrospective practices and the GoRetro tool is mostly preferable. Future research might examine different cooperative games and the use of retrospective practices/approaches in other agile practices like Extreme Programming, Kanban and Lean Development. We will also extend the scope of this study and cover other software industries as well.

VIII. Contribution And Future Work

In this research study retrospective practices and their impact of retrospective practices in Pakistani software industry. Main purpose of this study to process improvement of the software development.

Potential future research could investigate alternative collaborative gaming experiences and the use of retrospective practices/approaches in other agile practices like Extreme Programming, Kanban and Lean Development. We will also extend the scope of this study and cover other software industries as well.

References

[1] One, V. (2017). 11th annual state of agile survey. *Acessado em*, *6*.

[2] Boehm, B., Grunbacher, P., & Briggs, R. O. (2001). Developing groupware for requirements negotiation: lessons learned. *IEEE software*, *18*(3), 46-55.

[3] Campbell, J., Kurkovsky, S., Liew, C. W., & Tafliovich, A. (2016, February). Scrum and agile methods in software engineering courses. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 319-320).

[4] Przybyłek, A., Albecka, M., Springer, O., & Kowalski, W. (2022). Game-based Sprint retrospectives: multiple action research. *Empirical Software Engineering*, *27*(1), 1-56.

[5] Butt, S. A., Tariq, M. I., Jamal, T., Ali, A., Martinez, J. L. D., & De-La-Hoz-Franco, E. (2019). Predictive variables for agile development merging cloud computing services. *IEEE Access*, *7*, 99273-99282.

[6] Gaikwad, P. K., Jayakumar, C. T., Tilve, E., Bohra, N., Yu, W., & Spichkova, M. (2019). Voice-activated solutions for agile retrospective sessions. *Procedia Computer Science*, *159*, 2414-2423.

[7] Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, *22*(5), 30-39.

[8] Moe, N. B., Aurum, A., & Dybå, T. (2012). Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology*, *54*(8), 853-865.

[9] von Rosing, M., von Scheel, J., & Gill, A. Q. (2015). Applying Agile Principles to BPM.

[10] Jovanovic, M., Lalić, B., Mas, A., & Mesquida, A. L. (2015). The agile approach in industrial and software engineering project management. *Journal of Applied Engineering Science*, *13*(4).

[11] Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software development*, *9*(8), 28-35.

[12] Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of systems and software*, *85*(6), 1213-1221.

[13] Yilmaz, M., & O'Connor, R. V. (2016). Improving Social Aspects of the Software Development Process: Games, Gamification and Related Approaches J. UCS Special Issue. *Journal of Universal Computer Science*, *22*(12), 1487-1488.

[14] Stettina, C. J., Offerman, T., De Mooij, B., & Sidhu, I. (2018, June). Gaming for agility: using serious games to enable agile project & portfolio management capabilities in practice. In *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (pp. 1-9). IEEE.

[15] Akarsu, Z., Metin, Ö. O., Gungor, D., & Yilmaz, M. (2018, September). Towards a role playing game for exploring the roles in Scrum to improve collaboration problems. In *European Conference on Software Process Improvement* (pp. 254-264). Springer, Cham.

[16] Üsfekes, Ç., Tüzün, E., Yılmaz, M., Macit, Y., & Clarke, P. (2019). Auction-based serious game for bug tracking. *IET Software*, *13*(5), 386-392.

[17] Daylamani-Zad, D., Agius, H., & Angelides, M. C. (2020). Reflective agents for personalisation in collaborative games. *Artificial Intelligence Review*, *53*(1), 429-474.

[18] Adzic, G., Evans, D., & Roden, T. (2015). *Fifty Quick Ideas to Improve Your Tests*. Neuri Consulting LLP.

[19] Matthies, C., & Dobrigkeit, F. (2021, January). Experience vs Data: A Case for More Data-Informed Retrospective Activities. In *International Conference on Lean and Agile Software Development* (pp. 130-144). Springer, Cham.

[20] Scrum Alliance. 2018. *State of Scrum 2017-2018: Scaling and Agile Transformation*. Technical Report. 36 pages. http://info.Scrumalliance.org/State-of-Scrum-2017-18.html

[21] KNIBERG, H. (2015). Scrum and Xp from the Trenches 2nd Edition. 2. Sl: Lulu. com.

[22] CollabNet, V. (2019). 13th Annual State Of Agile Report. *CollabNet VersionOne, Alpharetta, GA, USA*.

[23] Matthies, C., Teusner, R., & Hesse, G. (2018, October). Beyond surveys: Analyzing software development artifacts to assess teaching efforts. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.

[24] Atlasian. https://www.atlassian.com/agile

[25] Atlasian.https://ww.atlassian.com/agile/manidesto

[26] Scrum.https://www.Scrum.org/resources/what-is-Scrum

[27] Schwaber, k., Sutherland, J. (2020) The Scrum Guide — The Definitive Guide to Scrum: The Rules of the Game..http://www.Scrumguides.org

[28] Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft press.

[29] Lehtinen, T. O., Itkonen, J., & Lassenius, C. (2017). Recurring opinions or productive improvements—what agile teams actually discuss in retrospectives. *Empirical Software Engineering*, *22*(5), 2409-2452.

[30] Ilyés, E. (2019, September). Create your own agile methodology for your research and development team. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 823-829). IEEE.

[31] Dingsøyr, T., Mikalsen, M., Solem, A., & Vestues, K. (2018, May). Learning in the large-an exploratory study of retrospectives in large-scale agile development. In *International Conference on Agile Software Development* (pp. 191-198). Springer, Cham.

[32] Mas, A., Poth, A., & Sasabe, S. (2018, September). SPI with retrospectives: a case study. In *European Conference on Software Process Improvement* (pp. 456-466). Springer, Cham.

[33] Küpper, S., Pfahl, D., Jürisoo, K., Diebold, P., Münch, J., & Kuhrmann, M. (2019). How has SPI changed in times of agile development? Results from a multi-method study. *Journal of Software: Evolution and Process*, *31*(11), e2182.

[34] Schwaber, K., Larsen, D., & Derby, E. (2006). Agile retrospectives: Making good teams great. *Agile Retrospectives*, 1-178.

[35] Matthies, C. (2020, October). Playing with your project data in Scrum retrospectives. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)* (pp. 113-115).

[36] McGonigal, J. (2011). *Reality is broken: Why games make us better and how they can change the world*. Penguin.

[37] Csikszentmihalyi, M., & Csikzentmihaly, M. (1990). *Flow: The psychology of optimal experience* (Vol. 1990). New York: Harper & Row.

[38] Sawyer, R. K. (2007). Group Genius: The Creative Power of Collaboration. Chap. 4.

[39] Von Neumann, J., & Morgenstern, O. (2007). Theory of games and economic behavior. In *Theory of games and economic behavior*. Princeton university press.

[40] Zagal, J. P., Rick, J., & Hsi, I. (2006). Collaborative games: Lessons learned from board games. *Simulation & gaming*, *37*(1), 24-40.

[41] Abt, C. C. (1987). *Serious games*. University press of America.

[42] Gonçalves, L., & Linders, B. (2015). *Getting value out of agile retrospectives: a toolbox of retrospective exercises*. Ben Linders Publishing.

[43] Przybylek, A., & Olszewski, M. K. (2016, September). Adopting collaborative games into Open Kanban. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 1539-1543). IEEE.

[44] Hohmann, L. (2006). *Innovation games: creating breakthrough products through collaborative play*. Pearson Education.

[45] Lin, L. H., Lin, W. H., Chen, C. Y., & Teng, Y. F. (2010, June). Playfulness and innovation—A multilevel study in individuals and organizations. In *2010 IEEE International Conference on Management of Innovation & Technology* (pp. 771-776). IEEE.

[46] Ghanbari, H., Similä, J., & Markkula, J. (2015). Utilizing online serious games to facilitate distributed requirements elicitation. *Journal of Systems and Software*, *109*, 32-49.

[47] Przybyłek, A., & Zakrzewski, M. (2018). Adopting collaborative games into agile requirements engineering.

[48] Santana, C., Queiroz, F., Vasconcelos, A., & Gusmão, C. (2015, August). Software process improvement in agile software development a systematic literature review. In *2015 41st Euromicro Conference on Software Engineering and Advanced Applications* (pp. 325-332). IEEE.

[49] Kua, P. (2012). The Retrospective Handbook: A guide for agile teams.

[50] Matthies, C., Dobrigkeit, F., & Ernst, A. (2019, September). Counteracting agile retrospective problems with retrospective activities. In *European Conference on Software Process Improvement* (pp. 532-545). Springer, Cham.

[51] Matthies, C., & Dobrigkeit, F. (2019). Towards empirically validated remedies for Scrum retrospective headaches. *arXiv preprint arXiv:1910.08763*.

[52] Jovanović, M., Mesquida, A. L., Radaković, N., & Mas, A. (2016). Agile retrospective games for different team development phases. *Journal of Universal Computer Science*, *22*(12), 1489-1508.

[53] Caroli, P., & Caetano, T. Fun retrospectives-activities and ideas for making agile retrospectives more engaging. Leanpub. com (2016).

[54] Przybyłek, A., & Kotecka, D. (2017, September). Making agile retrospectives more awesome. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 1211-1216). IEEE.

[55] Kotecka, D. Enhancing Scrum with collaborative games. *MSc thesis, Gdansk University of Technology*, 2016

[56] Adapt Methodology.https://adaptmethodology.com/starfish-exercise/

[57] GoRetro.https://www.goretro.ai/

[58] Sandoval-Alfaro, O. E., & Quintero-Meza, R. R. (2021, August). Application of Data Analytics Techniques for Decision Making in the Retrospective Stage of the Agile Scrum Methodology. In *2021 Mexican International Conference on Computer Science (ENC)* (pp. 1-8). IEEE.

[59] Kanwal, J., Basit, H. A., & Maqbool, O. (2018, March). Structural clones: An evolution perspective. In *2018 IEEE 12th International Workshop on Software Clones (IWSC)* (pp. 9-15)

[60] Kanwal, J., Inoue, K., & Maqbool, O. (2017, February). Refactoring patterns study in code clones during software evolution. In *2017 IEEE 11th International Workshop on Software Clones (IWSC)* (pp. 1-2).

[61] Kanwal, J., Maqbool, O., Basit, H. A., & Sindhu, M. A. (2019). Evolutionary perspective of structural clones in software. *IEEE Access*, *7*, 58720-58