





# Counteracting Agile Retrospective Problems with Retrospective Activities

Christoph Matthies<sup>(✉)</sup> , Franziska Dobrigkeit , and Alexander Ernst

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany  
{christoph.matthies, franziska.dobrigkeit, alexander.ernst}@hpi.de

**Abstract.** Retrospective meetings are a fundamental part of Agile software development methods. Effective retrospectives can positively impact teamwork, productivity, and work satisfaction. In this paper, we focus on problems that commonly occur during these meetings. To address them, we suggest retrospective activities that are already used in practice. These activities provide structure and guide the team through the meeting. We created a mapping between a selection of these activities and the problems they attempt to solve. We evaluated the created mapping through multiple case studies with software development teams in educational and professional contexts. Our results verify the existence of a relationship between specific retrospective activities and connected retrospective problems. Furthermore, using observational studies we could verify parts of the created problem-activity mapping.

**Keywords:** Software process improvement · SPI · Agile software development · Retrospective meeting

## 1 Introduction

Agile methods have firmly established themselves in professional software development [24, 31, 42]. Software process improvement approaches, e.g. the Scrum Sprint retrospective, are integral features of Agile development methods. The State of Agile survey lists retrospectives as the third most popular Agile technique employed in the industry, conducted by 85% of survey participants [44]. According to the 2018 report by the Scrum Alliance, 81% of the participants followed the Scrum Guide [41], conducting retrospectives after each Sprint [42]. Retrospectives provide time and space to discuss and improve a team's development process, leading to increased performance and enjoyment during the following iteration [35, 40]. These meetings allow implementing the principles “motivate all people involved” and “create a learning organization” of the Software Process Improvement (SPI) manifesto [37]. However, retrospective meetings face their own set of challenges. Process facilitators and coaches have identified problems that frequently emerge during these meetings, hindering teams in realizing their retrospective's full potential, e.g. participants not speaking up [1, 39].

Several researchers and practitioners have therefore proposed activities to structure retrospectives [6, 9, 18, 26]. These activities, or “games” [20, 21, 34] “are timeboxed processes that [...] help your team think together” [9]. They support retrospectives by encouraging equal participation and the exploration of new perspectives [9, 22]. While the proposed activities address certain issues of retrospectives, explicit assignments of problems to activities currently only exist in a few cases, which have been identified as valuable for practitioners [22].

In this paper, we tackle this lack of guidance by proposing a mapping between frequently identified retrospective problems and activities, which can be employed by practitioners to select a fitting activity for a diagnosed problem. Our initial mapping is based on existing problems and activities commonly found in Agile literature and mentioned in online practitioner resources. It was evaluated in the context of case studies with software teams of students and professionals employing the Scrum methodology. The results indicate that while activities were mostly perceived as having a positive impact and addressed retrospective problems, some activities also led to additional issues.

## 2 Agile Retrospectives

Schwaber et al. define a retrospective as an “opportunity for the team to inspect itself” [41]. During these meetings the project team discusses a finished project or iteration, focusing on successes as well as encountered challenges and defines steps to improve their process in the future. Retrospectives can help establish the principles outlined in the SPI manifesto, by providing space to talk and to “ensure all parties understand and agree on process” [37]. The output of retrospective meetings is a list of action items, including the next steps necessary to proceed and address the identified issues [22]. Retrospectives originally appeared in the form of postmortem meetings at the end of a project to learn from success or failures. Kerth’s list of basic retrospective questions [23], used as a foundation for every retrospective, are still relevant today and are still being taught [33, 36]. Agile retrospectives form an integral part of the Scrum framework [41]. The importance of retrospectives for the Agile community is reflected in the last principle of the Agile Manifesto: “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.” [15].

While the basic concept and setup of retrospectives are intuitive, research on retrospective problems shows that executing successful retrospectives is challenging. Kua [26] and Rubin [39] collected agile retrospective smells, such as *All Talk, No Action* or the *Blame Game*. These authors were also the only ones to also suggest solutions to the identified problems, listing active interventions by the retrospective facilitator, e.g. asking specific questions at opportune moments.

## 3 Research Methodology and Context

Our aim in this research is to provide guidance to facilitators in choosing activities which counteract identified issues in retrospectives of their teams. The following research question guides our work:

*RQ.* Which activities can be applied to address common retrospective problems? Our research approach is based on the following hypotheses:

- H1.** Existing Agile retrospective activities already address retrospective problems without explicitly mentioning so.
- H2.** Scrum Masters (SMs) can address problems that occur in their teams' retrospectives with problem-specific activities.

We followed the following steps: (i) identify and collect retrospective problems and activities from research literature and online resources, (ii) map retrospective activities to the problems they address, (iii) conduct case studies in Agile teams to evaluate the real-world influence of activities on the corresponding problems.

### 3.1 Collecting Common Problems of Retrospectives

We collected known retrospective problems from the related literature, but also explicitly included problems listed on popular practitioner websites, as these are the resources most used by and most relevant to practitioners [4], much more so than research papers [10]. These resources, being easy to modify, also have the advantage of being most up to date. We based our collection of retrospective problems on the following sources: The books “The Retrospective Handbook” [26] by Kua, Rubin’s “Essential Scrum” [39], and Loeffler’s “Improving Agile Retrospectives” [30] discuss and list retrospective problems. Amin’s article on the “Do’s and Don’ts of Agile Retrospectives” [1], published by the Scrum Alliance, is of relevance especially due to the publishing venue and the active community. Gonçalves’ blog post on the “9 Deadly Agile Retrospectives Antipatterns” [17] contains a curated list of problems derived from practice. The *Agile Retrospective Wiki* entry on “Common Ailments and Cures”, created by Bowley and Linders, lists common retrospective problems and has been viewed more than 100,000 times [5].

The resulting collection of retrospective problems was cleaned of duplicates. Furthermore, problems which could not be addressed by applying a retrospective activity were excluded from our analysis. Removed problems included those where (i) solution approaches were inherently given, e.g. not capturing action items, (ii) where organizational solutions outside of retrospective structure were required, e.g. too little time, or (iii) where problems violated fundamental Agile principles, e.g. automatically assigning action items.

### 3.2 Collecting Common Retrospective Activities

We based our approach for gathering retrospective activities on recent previous collection efforts by Jovanović et al. [22] and Loeffler [30]. Retrospective activities were collected from the following sources. The books “Agile Retrospectives: Making Good Teams Great” by Derby and Larsen [9] as well as “Innovation Games” by Hohmann [20] have had major impacts on the Agile community. In recent years Caroli and Caetano (“Fun Retrospectives” [6]), Gonçalves and

Linders (“Getting Value out of Agile Retrospectives” [18]), as well as Krivitsky (“Agile Retrospective Kickstarter” [25]) and Kua (“The Retrospective Handbook” [26]) have continued this work. These sources were also used in previous work to map collected activities to team phases [22].

As with retrospective problems, we extended our collection by also inspecting relevant practitioner websites and continued the deduplication of activities as described by Jovanovic et al. [22]. Diana Larsen’s *Partnerships and Possibilities Blog* [28] was included due to her high regard in the Agile community. The *Retromat* is a practitioner website as well as a book with a broad collection of activities [3]. It is a popular tool built on Derby and Larsen’s phase model [9], which maps 129 activities to their respective phases within a retrospective meeting [3], and is used to plan individualized retrospectives [36]. We furthermore inspected the *Agile Retrospective Resource Wiki*, a “resource for sharing retrospective plans, tips & tricks, tools and ideas”, collaboratively edited by Agile practitioners. Specifically, we reviewed the entry on “Retrospective Plans”, as this collection of activities is of high interest to retrospective facilitators, having been accessed over 800,000 times [45].

### 3.3 Problem-Activity Mapping Approach

As we evaluated the created mapping in fourteen retrospectives and aimed to test individual activities repeatedly with multiple teams, the number of applied activities was constrained to ten. We applied the following list of criteria for selecting activities to be employed in teams: (i) Activities should be drawn from every source listed in this section in order to obtain variation in activities, (ii) the activity must address at least one issue listed in our collection of relevant retrospective problems, (iii) the activity must be mentioned by a minimum of two sources to ensure relevance for the Agile community.

For each activity matching these criteria, we compiled the descriptions and explanations given in the primary sources. Based on discussion and qualitative analyses of these characterizations, we matched the described issues to the collected solution approaches on a consensus basis between three involved researchers. The rationales for these decisions are described in Sect. 4.2.

### 3.4 Evaluation Approach

To evaluate our created problem-activity mapping, we conducted case studies as described by Flick [14]. Overall, we observed nineteen retrospectives: sixteen in a university context and three in a setting with professional software development teams. While five retrospectives were solely observed, in fourteen retrospectives we proposed activities, influencing the way the meetings were organized.

**Retrospective Observations.** During the observations, we took part as “Observer as Participant” [16]. Following this type of observation, subjects were aware of the researcher in the room and were aware of the research goals. We

observed twelve retrospective meetings of four student Scrum teams in the context of our software engineering undergraduate capstone course in the winter semester of 2016/17. The learning goals of this hands-on project course are the application of the Scrum process and Agile best practices within a multi-team development project. We also observed a Scrum team of a startup company in the mobile gaming sector. The startup works in two-week sprints and teams conduct retrospectives after each sprint. Furthermore, we observed two retrospectives of software developers teams of an established global player in the software industry. Their sprint length was two weeks and they conducted one retrospective every sprint. For the facilitated retrospectives, we proposed activities to SMs to address problems which appeared in the team's previous retrospective. These suggestions were made within a face to face meeting in which we explained the specific activity and its purpose to the SMs. The problems to be addressed were identified either through a structured interview with the SM or if possible by observing the previous retrospective.

**Surveys.** After the retrospective, we distributed a questionnaire to team members, capturing the perceptions of team members regarding the effects of the applied activity. We asked three different types of questions: Linear scale and multiple-choice questions which belong to closed questions, and open questions. Therefore, the questionnaire is a mixture of closed and open formats as defined by Leung [29]. The open questions were analyzed by means of coding as defined by Flick [14]. We labeled the free text with categories and looked for similarities and differences.

**Expert Interviews.** To evaluate the applicability of activities we conducted six semi-structured interviews [7] with participating SMs. This approach produced comparable data while preserving the interviewee's freedom to express their views [7]. The interviews were conducted at the end of the case studies. During the interview, we took notes that were coded and clustered in a following step [14]. The SMs involved in the case study applied the suggested activities and experienced their effects at first-hand. They, therefore, represent experts to evaluate research hypothesis H2.

## 4 Problem-Activity Mapping

### 4.1 Collected Retrospective Problems

Based on the sources identified in Sect. 3.1, we created a collection of retrospective problems. In the following we list the collected problems relevant to our research, which can be addressed by the use of retrospective activities:

- **All Talk–No Action** – failing to decide on action items defining the next steps and therefore having no clear path for improvement.
- **Too Repetitive** – repetitive retrospectives lead to fatigue and boredom, thus negatively affecting outcomes and team motivation.

- **No Preparation** – not thoroughly preparing a meeting, leaving participants feeling lost and their time not being valued.
- **Blame Game** – when a team is pointing fingers and assigning blame to team members instead of constructively searching for solutions.
- **Not Speaking Up** – team members are reluctant to openly address their feelings and perceptions or known problem in the group.
- **Taking It Personally** – team members feel hurt by the statements of others, leading to a charged and adversarial atmosphere.
- **Group Think** – the team values harmony and consensus higher than a critical discussion, which leads to fewer improvement possibilities.
- **Focus on Negatives** – a team losing its perception of positives from the past sprint due to a focus on negatives leading to negative team spirit.
- **Complain Game** – Participants viewing the meeting as a chance to complain and unload regarding the past instead of a chance to improve.

## 4.2 Selected Retrospective Activities

In the following, we describe the activities selected to be part of our mapping in more detail and explain which problems they are mapped to and why. Table 1 shows the resulting problem-activity mapping.

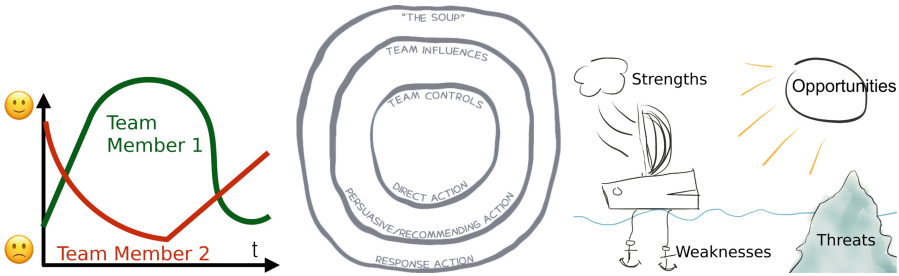
**Sailboat** [11, 18, 20, 25]. The Sailboat is a popular retrospective activity. The main idea is to think about the past sprint with a sailboat metaphor, see Fig. 1. Four elements influence the boat: Wind (positives), anchors (negatives), sun (opportunities) and icebergs (dangers). This describes the strengths and weaknesses of the past sprint and the opportunities and threats that lie ahead. After a brainstorming session, the team members share their items and categorize them, e.g. by sticking notes to a whiteboard.

**Futurespective** [6, 20]. The team imagines they have traveled to the future and are conducting their retrospective at this point in time, e.g. at the end of the next sprint. Based on this, the team brainstorms what made the previous sprint successful and defines action items. Because of the nature of this method, the team does not look back on its past, but only looks into the future.

**Open the Box** [6]. The Open the Box activity facilitates brainstorming with the help of a box metaphor. The box contains all actions that are performed by the team. When brainstorming, the team discusses three questions: “Which new actions should be put into the box?”, “Which actions should stay in the box?” and “Which actions should be thrown out?”

**Peaks and Valleys Timeline** [6]. Participants are motivated to communicate and to relate different views on similar events. Participants plot the development of their feelings during the last sprint in a shared mood graph, see Fig. 1.

**Guess Who** [2, 11, 38]. Guess Who is an activity, supporting the team’s empathy. Every participant answers the standard retrospective questions (see Sect. 2) from the eyes of another team member. After sharing the insights, the team



**Fig. 1.** Visualizations used in “Peaks and Valleys Timeline”, “Circles and Soup”, and “Sailboat” retrospective activities (from left to right).

guesses whose point of view the team member tried to take. The represented team member can add or decline the mentioned issues afterwards.

**Circles and Soup** [3,27]. While sharing the discovered problems, the team discusses which level of influence the team has, illustrated in Fig. 1. There are three levels: Control (issues can be addressed within the team), Influence (people outside the team are needed to address the issues) and Soup (addressing these issues is not in the influence of the team).

**Story Oscars** [43]. This is a short activity, meant to start the retrospective. Team members can nominate stories for several categories, such as the most exciting or the most challenging story. After suggesting different stories, the team votes which story gets the Oscar in the specific category.

**Tweet My Sprint** [19]. The team “tweets” about the past iteration. This is done offline and on sticky notes. The team is also invited to put hashtags, emoticons, attached pictures, and @usernames on the notes. After each team member has had time to create tweets, the notes are shared and arranged in a timeline. The team members can then retweet or reply.

**6 Thinking Hats** [5,8,11]. This activity allows thinking of the past iteration from different viewpoints. The team puts on different metaphorical hats, i.e. ways of thinking about the past iteration, such as positive or negative framing.

**Five Whys** [3,9,18,22]. This tool is meant to support a root cause analysis for a difficult underlying problem that hinders the team to do their work effectively. The team continuously repeats the question “Why?” until they are able to identify the lowest cause - usually after four to seven levels of this exercise.

### 4.3 Retrospective Activities Proposed by SMs During Case Studies

In meetings with SMs on the upcoming retrospective, we adapted to the team’s specific situation. The research design was modified to accommodate new activities requested by SMs for their teams, which are described in the following.

**Oscars for Team Members.** This activity, proposed by an SM, is based on the Story Oscars activity. Instead of giving awards to stories, team members award each other for their past performance. The team members note an award category, such as “Team spirit booster”, and select a recipient.

**Collective Painting.** This activity was suggested by an SM during an observational case study. The team collectively paints their impressions of the past sprint, without talking. While painting and observing the others work, a common understanding of the past sprint is reached.

**Reverse Brainstorming [13,46].** Reverse Brainstorming is a technique taken from Design Thinking approaches. However, it can also be used to generate action items within retrospectives. The team brainstorms ideas for the prompt “How can we make the next sprint the worst ever?”. The team then inverts each produced action item and defines the next steps.

#### 4.4 Constructing the Problem-Activity Mapping

When creating the problem activity mapping we generated hypotheses regarding which methods tackle the given problems. First, we took notice of the fact that several of the collected methods prescribe an inherent structure by making use of a form of brainstorming followed by a sharing of ideas and thoughts and ending with a discussion, e.g. *Open the Box*. These approaches help SMs prepare a retrospective with limited effort, as the phases of collecting issues and discussion are covered. The SM is then free to focus on facilitating discussions and creating specific action items. Accordingly, we believe that such methods with inherent structure can address the problem of *No Preparation*. Secondly, we note that several of these structured methods make use of silent brainstorming followed by the sharing of results, providing both thinking time and time to speak to every participant, e.g. *Circles and Soup*.

Introducing individual, silent thinking time can encourage team members to reflect, while sharing carefully considered points enables voicing thoughts, helping to address the problem of *Not Speaking Up*. Similarly, we found that some methods specifically investigate personal feelings, e.g. the *Peaks and Valleys Timeline*, which deals with participants’ emotional journey throughout the sprint. Being prompted to explicitly share personal thoughts can support team members in sharing their experiences and thus also addresses the problem of *Not Speaking Up*. In the following, we describe the reasoning for the created problem-activity associations. The complete mapping is depicted in Table 1.

**Sailboat [11,18,20,25].** This activity should address *No Preparation*, by providing an idea of how to search for topics and how to cluster them. It should also address *Not Speaking Up*, due to the silent brainstorming and sharing sessions. Additionally, the Wind and Sun categories emphasize the positives of the past and future opportunities thus should address *Focus on Negatives*.

**Futurespective [6,20].** The retrospective meeting can be replaced with a “Futurespective”, as they share the same goals of future process improvement.



**Table 1.** Problem-activity mapping.

Activity	Addressed problems
Sailboat	No Preparation, Not Speaking Up, Focus on Negatives
Futurespective	Too Repetitive, No Preparation, Focus on Negatives
P+V Timeline	No Preparation, Blame Game, Not Speaking Up
Open the Box	All Talk-No Action, No Preparation, Not Speaking Up, Focus on Negatives
Story Oscars	Not Speaking Up, Focus on Negatives
Guess Who	Blame Game, Taking It Personally
Circles + Soup	Too Repetitive, Focus on Negatives, Complain Game
6 Thinking Hats	All Action No Talk, Group Think
Five Whys	All Action No Talk
Tweet my Sprint	Too Repetitive, No Preparation, Not Speaking Up
Guess Who, P+V Timeline	Blame Game, Taking It Personally
Collective Painting	Conversation Too Controlled, Too Repetitive, Not Speaking Up
Team Members Oscars	Blame Game, Not Speaking Up, Taking It Personally, Focus on Negatives
Reverse Brainstorming	All Talk-No Action, Too Repetitive

This variation can help address *Too Repetitive*. Additionally, the activity can reduce *Focus on Negatives*, as negative, past experiences not discussed. Finally, the inherent structure of the method can address *No Preparation*.

**Open the Box** [6]. Due to its structure, this activity encourages defining action items at an early stage and is suitable to address *All Talk-No Action* and *No Preparation*. Similar to the Sailboat, Open the Box gives every team member time to reflect in silence and provides structure by providing three categories for outcomes, helping to address *Not Speaking Up*. The category “Keep in the box” provides a positive outlook, tackling the issue of *Focus on Negatives*.

**Peaks and Valleys Timeline** [6]. This method focuses on personal feelings and can address *Blame Game*. As it provides a detailed structure in which to think and share, it can help counteract *Not Speaking Up* and *No Preparation*.

**Guess Who** [2, 11, 38]. This activity supports team spirit and empathy, as team members try to see the past from a different point of view. Thus the activity can address the problems *Blame Game* and *People taking things Personally*.

**Circles and Soup** [3, 27]. The goal of the activity to focus on the problems the team can address and to accept the ones that are out of the team’s influence, thus it can address *Complain Game* and *Focus on Negatives*. Furthermore, the activity takes a different approach to clustering and thus addresses *Too Repetitive*.

**Story Oscars** [43]. The intent of this activity is to motivate discussions among participants. Thus, it can address *Not Speaking Up*. Furthermore, the use of positive categories is encouraged the activity can help reduce *Focus on Negatives*.

**Tweet My Sprint** [19]. This activity differs from more common activities and therefore should address *Too Repetitive*. Additionally, it allows communication without speaking and can thus address the problem *Not Speaking Up*. Furthermore, the inherent structure of the activity can help tackle *No Preparation*.

**6 Thinking Hats** [5, 8, 11]. Looking at a topic from six different viewpoints helps in not making decisions too fast. Thus, this method can address the problem *All Action No Talk*. Additionally, being forced to look at a problem from varying viewpoints can help team members overcome *Group Think*.

**Five Whys** [3, 9, 18, 22]. Five Whys is suitable to address *All Action No Talk*, as the team takes a step back and analyzes the root cause of the problem, instead of agreeing on action items too fast.

**Oscars for Team Members.** This activity focuses on recognizing positive, personal performance and can, therefore, address *Focus on Negatives*, *Blame Game*, *People taking things personally* and *Not Speaking Up*.

**Collective Painting.** This method can address *Not Speaking Up* and *Conversation Too Controlled*, as every team members can individually express their perceptions through artistic means. Additionally, it is an unusual activity, differing in structure and means of expression, and can, therefore, address the problem of retrospectives being *Too Repetitive*.

**Reverse Brainstorming** [13, 46]. This activity can address the problem *All Talk–No Action*, as the team is encouraged to produce action items. The principle of finding answers to the exact opposite of the original question varies drastically from more standard exercises, also addressing *Too Repetitive*.

## 5 Evaluation Results

During the retrospectives we observed the problems: *No Preparation*, *Not Speaking Up*, *All Talk–No Action*, and *Too Repetitive*. However, we could not observe every problem from our collection of relevant problems during the retrospectives we attended. Table 2 displays the activities we used during the fourteen retrospectives we helped facilitate along with the reason we chose the activity. Overall our observations and the answers to our questionnaire supported most of the problem – activity pairs we tested and the participants generally liked the activities they had used. Accordingly, we found that methods with an inherent structure (1<sup>st</sup> brainstorm, 2<sup>nd</sup> share, 3<sup>rd</sup> discuss,...), e.g. Sailboat, helped to prepare a retrospective with limited time and effort and thus addressed the problem *No Preparation*. Similarly, we found evidence that activities which provide silent thinking time and ask for personal feelings, e.g. Sailboat or Peaks and Valleys Timeline support team members in sharing their perceptions and thus address the problem *Not Speaking Up*.

In two cases we could not confirm our hypothesis that the activity addressed the problem: In the case of Tweet my Sprint the students reported very differently on its ability to create an atmosphere in which it is easy to address problems. We believe this might be due to the short format of tweets which does not allow for enough room when problems need to be analyzed and discussed in more detail. Additionally, the activity Reverse Brainstorming only minimally improved the number of topics discussed within the team, so that we could not confirm that this activity addresses *All Talk–No Action*.

**Table 2.** Activities used in the case studies and number of teams that were observed.

Activity	Reasons	Teams
Sailboat	No Preparation, Not Speaking Up, Diversification	4
Peaks + Valley Timeline	No Preparation, Not Speaking Up, Diversification	3
Open The Box	All Talk–No Action, No Preparation	2
Circles and Soup	Diversification	2
Team Member Oscars	Focus on Negatives	1
Reverse Brainstorming	All Talk–No Action	1
Futurespective	No Preparation	1
Collective Painting	Diversification	1
Tweet my Sprint	Not Speaking Up	1
Story Oscars	Diversification, Project Wrap-Up	1

The feedback from the SMs during the interviews was positive for the majority of the activities. The SMs stated that they could imagine using the activities again and understood their use cases. In particular, the SMs using Circles and Soups mentioned that it is useful in cases of process changes and helps the team to understand their influence on such changes. Overall, the SMs were able to use the activities and were able to transfer them to their teams in all observed cases.

## 6 Conclusions

We were able to identify multiple of the collected retrospective problems in retrospectives of teams from both professional and educational backgrounds. The results of the case studies indicate a positive effect of six retrospective activities on four retrospective problems. These results are the first steps towards an empirically sound mapping between retrospective activities and retrospective problems. Our results lend support to hypothesis *H1*: “Existing Agile retrospective activities already address retrospective problems without explicitly mentioning so”. This research showed connections between several common retrospective activities and retrospective problems collected from literature and practitioner websites. In the observational case studies of multiple teams, we introduced process facilitators to our problem-activity mapping and the selected retrospective activities. Scrum Masters were able to successfully introduce these activities into their regular team meetings when problems during retrospectives were identified. This confirms research hypothesis *H2* concerning SMs applying activities successfully. The problem which was most often addressed by the use of activities was *No Preparation*. We were able to confirm a positive effect on this problem for all of the conducted activities. This is in line with Derby’s definition of the goals of retrospective activities, i.e. to “provide structure to help the team think together” [9]. The prepared agenda of activities supports reduces the planning effort. However, the provided level of structure differs between the activities. Therefore, some activities address this problem more effectively than others.

The retrospective meeting is an important part of Agile development and related processes [12]. Our contributions include collecting and deduplicating

retrospective problems as well as verifying these issues in expert interviews. Furthermore, in case studies of educational and professional backgrounds, we observed a subset of these problems in retrospectives of teams. We have collected activities from literature and practitioner websites, which can address these problems. The resulting mapping is a first step in tackling the lack of connection between research on common retrospective issues and work on retrospective activities [22]. Future work will focus on refining this mapping and making it more accessible for practitioners, e.g. through websites or software solutions [32].

Our work contributes to the awareness of how common retrospective problems can be addressed by the use of already known activities. We believe this can help to increase the effectiveness of retrospectives and to improve teamwork, work satisfaction, quality of work and productivity [18] of the teams employing such a mapping in their retrospectives.

## References

1. Amin, H.: Do's and don'ts of agile retrospectives (2014). <https://www.scrumalliance.org/community/articles/2014/july/dos-and-don-ts-of-agile-retrospectives>. Accessed 02 Jan 2018
2. Atherton, R.: Guess Who? (2016). [http://retrospectivewiki.org/index.php?title=Guess\\_Who](http://retrospectivewiki.org/index.php?title=Guess_Who) . Accessed 16 Jan 2018
3. Baldauf, C.: Retromat - Run Great Agile Retrospectives!. Leanpub, Layton (2018)
4. Beecham, S., O'Leary, P., Baker, S., Richardson, I., Noll, J.: Making software engineering research relevant. *Computer* **47**(4), 80–83 (2014)
5. Bowley, R., Linders, B.: Common ailments & cures (2018). [http://retrospectivewiki.org/index.php?title=Common\\_ailments..](http://retrospectivewiki.org/index.php?title=Common_ailments..) Accessed 11 Jan 2019
6. Caroli, P., Caetano, T.: Fun Retrospectives - Activities and Ideas for Making Agile Retrospectives More Engaging. Leanpub, Layton (2015)
7. Cohen, D., Crabtree, B.: Qualitative research guidelines project (2006). <http://www.qualres.org/>
8. De Bono, E.: Six Thinking Hats. Penguin, London (2017)
9. Derby, E., Larsen, D.: Agile Retrospectives: Making Good Teams Great. Pragmatic Bookshelf Series, Pragmatic Bookshelf (2006)
10. Devanbu, P., Zimmermann, T., Bird, C.: Belief & evidence in empirical software engineering. In: Proceedings of the 38th International Conference on Software Engineering - ICSE 2016. pp. 108–119. ACM Press, New York (2016)
11. Dieu, L.: Stages of agile retrospective (2016). <https://apiumtech.com/blog/5-stages-of-agile-retrospective-and-7-popular-techniques/>. Accessed 16 Jan 2019
12. Dobrigkeit, F., de Paula, D., et al.: The best of three worlds-the creation of innodev a software development approach that integrates design thinking, scrum and lean startup. In: DS 87–8 Proceedings of the 21st International Conference on Engineering Design, ICED 2017, vol. 8, pp. 319–328 (2017)
13. Evans, N.: Destroying collaboration and knowledge sharing in the workplace: a reverse brainstorming approach. *Knowl. Manag. Res. Pract.* **10**(2), 175–187 (2012)
14. Flick, U.: An Introduction to Qualitative Research. Sage Publications Limited, Thousand Oaks (2009)
15. Fowler, M., Highsmith, J.: The agile manifesto. *Software Development* (2001)

16. Gold, R.L.: Roles in sociological field observations. *Soc. Forces* **36**, 217–223 (1958)
17. Gonçalves, L.: 9 deadly agile retrospectives antipatterns every scrum master must avoid (2017). <https://luis-goncalves.com/agile-retrospectives-antipatterns/>. Accessed 31 Jan 2019
18. Gonçalves, L., Linders, B.: Getting Value out of Agile Retrospectives - A Toolbox of Retrospective Exercises. Lulu. com (2014)
19. Guest, T.: #tweetmysprint (2019). <https://retromat.org/en/?id=97>. Accessed 16 Jan 2019
20. Hohmann, L.: *Innovation Games: Creating Breakthrough Products Through Collaborative Play*. Pearson Education, London (2006)
21. Jovanovic, M., Mesquida, A.-L., Mas, A.: Process improvement with retrospective gaming in agile software development. *Systems, Software and Services Process Improvement. CCIS*, vol. 543, pp. 287–294. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24647-5\\_23](https://doi.org/10.1007/978-3-319-24647-5_23)
22. Jovanović, M., Mesquida, A.L., Radaković, N., Mas, A.: Agile retrospective games for different team development phases. *J. Univ. Comput. Sci.* **22**(12), 1489–1508 (2016)
23. Kerth, N.L.: The ritual of retrospectives: how to maximize group learning by understanding past projects. *Softw. Test. Qual. Eng.* **2**(5), 53–57 (2000)
24. Komus, A., Kuberg, M.: Abschlussbericht: Status quo agile 2016/2017. Studie über Erfolg und Anwendungsformen von agilen Methoden (2017)
25. Krivitsky, A.: *Agile Retrospective Kickstarter*. Leanpub, Layton (2015)
26. Kua, P.: *The Retrospective Handbook: A guide for agile teams*. CreateSpace Independent Publishing Platform, Scotts Valley (2013)
27. Larsen, D.: Circles and soup (2010). <https://www.futureworksconsulting.com/blog/2010/07/26/circles-and-soup/>. Accessed 16 Jan 2019
28. Larsen, D., Shore, J.: Partnerships and possibilities blog (2019). <https://www.futureworksconsulting.com/blog/>. Accessed 16 Jan 2019
29. Leung, W.C.: How to design a questionnaire. *Stud. BMJ* **9**, 171–216 (2001)
30. Loeffler, M.: *Improving Agile Retrospectives: Helping Teams Become More Efficient*. Addison-Wesley, Boston (2017)
31. Matthies, C.: Feedback in scrum: data-informed retrospectives. In: *Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings, ICSE 2019*, pp. 198–201. IEEE Press (2019)
32. Matthies, C., Dobrigkeit, F., Hesse, G.: An additional set of (automated) eyes: chatbots for agile retrospectives. In: *1st International Workshop on Bots in Software Engineering (BotSE)* (2019)
33. Matthies, C., Teusner, R., Hesse, G.: Beyond surveys: analyzing software development artifacts to assess teaching efforts. In: *2018 IEEE Frontiers in Education Conference (FIE)*, pp. 1–9. IEEE (2018)
34. Mesquida, A.-L., Jovanovic, M., Mas, A.: Process improving by playing: implementing best practices through business games. In: Kreiner, C., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) *EuroSPI 2016. CCIS*, vol. 633, pp. 225–233. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-44817-6\\_18](https://doi.org/10.1007/978-3-319-44817-6_18)
35. Mesquida, A.-L., Karać, J., Jovanović, M., Mas, A.: A game toolbox for process improvement in agile teams. In: Stolf, J., Stolf, S., O'Connor, R.V., Messnarz, R. (eds.) *EuroSPI 2017. CCIS*, vol. 748, pp. 302–309. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-64218-5\\_25](https://doi.org/10.1007/978-3-319-64218-5_25)
36. Northwood, C.: *Planning Your Work*, pp. 11–46. Apress, Berkeley (2018)
37. Pries-Heje, J., Johansen, J.: *SPI Manifesto. European system & software process improvement and innovation* (2010)

38. Remarczyk, M.: “Guess Who?” (2015). <http://richardatherton.net/insights/guess-who-a-retrospective-from-a-different-perspective/>. Accessed: 16 Jan 2019
39. Rubin, K.S.: Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley, Boston (2012)
40. Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, Redmond (2004)
41. Schwaber, K., Sutherland, J.: The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game. Technical report (2017)
42. Scrum Alliance: The State of Scrum Report 2017 / 2018. Technical report (2018)
43. Todorov, M.: Story Oscars (2019). <https://retromat.org/en/?id=54>. Accessed 16 Jan 2019
44. VersionOne Inc.: The 12th Annual State of Agile Report. Technical report (2018)
45. Wiki Contributors: Statistics (2019). <http://retrospectivewiki.org/index.php?title=Special:Statistics>. Accessed 31 Jan 2019
46. Wilson, C.E.: Brainstorming pitfalls and best practices. ACM Interact. **13**(5), 50 (2006)