

Don't Break the Build: Developing A Scrum Retrospective Game

David G. Marshburn
Cleveland State University
d.marshburn@csuohio.edu

JP Sieck
Blue Ridge Informatics, LLC
jsieck@blueridgeinformatics.com

Abstract

This study discusses the design and observed play of a game-based Scrum retrospective. The game builds on the existing wealth of retrospective activities but adds in actual game play. The game is created in such a way as to satisfy the definition of a game and includes a win/loss state uncommon within typical retrospective activities. Leveraging existing design paradigms, the game looks to capitalize on the reported benefits of using games in team building and learning environments. The game fulfills the goals of a Scrum retrospective for the team to inspect and adapt processes by guiding the team in focused discussion regarding their performance and observations during the proceeding Sprint. The study provides an overview of the game design and mechanics and provides observations and results from post-game questionnaires. Finally, the study proposes changes to the game based on results of the observations and discusses future research possibilities.

1. Introduction

Retrospectives are one of the four ceremonies required within the Scrum framework [1] and serve to provide a development team a chance to “inspect itself and create a plan for improvements to be enacted during the next Sprint” [1, p. 14]. Schwaber and Sutherland state that retrospectives should allow teams to improve “process and practices to make it more effective and enjoyable for the next Sprint . . . [and to] increase product quality by improving processes” [1, p. 14].

In practice, teams generally follow either an activity-based or non-activity-based paradigm within retrospectives. Activity-based retrospectives use structured processes to help elicit feedback from teams whereas non-activity-based retrospectives rely on more proactive participation from team members. Derby and Larsen [2] recommend that teams should perform activity-based retrospectives to encourage

and maintain team engagement and should frequently change the retrospective activities. Przybylek and Kotecka [3] further Derby and Larsen’s [2] recommendation by providing anecdotal evidence that activity-based retrospectives are more effective than non-activity-based retrospectives. Recently, Marshburn [4] hypothesizes that game-based retrospectives are more effective than both activity-based and non-activity-based retrospectives. Marshburn [4] notes, however, that as there are currently no effectiveness measures for retrospectives these claims cannot be evaluated quantitatively. To resolve this issue, Marshburn [4] proposes to develop quantitative retrospective measures and to conduct an experiment to determine if there are differences between game-based, activity-based, and non-activity-based retrospectives.

The objective of this study is to systematically develop a game to be used in Scrum retrospectives that adheres to the definition of a game and that emphasizes the game components necessary to engage participants while enabling the improvement of processes and team dynamics within a Scrum team. The current study focuses only on the development of the Scrum retrospective game, but the resultant game can be used to support the research proposed by Marshburn [4].

The remainder of the paper is organized as follows. First, a brief literature review that addresses retrospectives, games, gamification, and game design is presented. This is followed by an overview of the research methodology which encompasses the development of the Scrum retrospective game *Don't Break the Build*. The research then reviews the observed play of *Don't Break the Build* and provides conclusions and recommendations for continued research, development, and testing of the game.

2. Literature Review

2.1. Retrospectives

Within the Scrum framework, retrospectives are:

a special meeting where the team gathers after completing an increment of work to inspect and adapt their methods and teamwork. Retrospectives enable whole-team learning, act as catalysts for change, and generate action. . . retrospectives focus not only on the development process, but on the team and team issues - [2, p. xi].

Derby and Larsen [2] have used retrospectives with Scrum teams to improve (i) productivity, (ii) capability, (iii) quality, and (iv) capacity. Retrospectives are a required ceremony within Scrum [1] and a means through which Scrum development teams can meet the Agile principle calling for continual process improvement [5]. Furthermore, retrospectives serve to respond to the decades old practice of project post-mortems [6], but do so in an iterative process during the development cycle in order to afford a direct effect on the current project.

Modern Agile retrospectives have improved and adapted significantly since the post-mortem days of the 1990's [7] when post-mortems were rigid, top-down processes held at the end of projects and which often did not result in meaningful improvements that could be used as lessons learned for other projects [8]. Not only are Scrum retrospectives held regularly throughout the development cycle, they are team-focused and serve as a means through which the team can inspect and adapt their own practices in real-time with immediate results [1].

For many Scrum teams, retrospectives are non-activity-based. In non-activity-based retrospectives, team members gather and discuss the preceding Sprint, with team members self-guiding the discussion. Non-activity-based retrospectives rely on the team members to proactively identify and discuss issues in an ad-hoc manner.

Activity-based retrospectives, on the other hand, foster team discussion and collaboration through the use of activities. Retrospective activities are often theme based and serve as a guide to help team members in identifying both good and bad aspects of the preceding Sprint. An example of a retrospective activity is the Sail Boat retrospective (this is a common retrospective activity found on many internet sites such as [9]) in which a metaphor is used to equate a team's effectiveness to a sail boat traveling down a river. In the Sail Boat retrospective, the Scrum Master draws a sail boat, complete with anchor, waves, the wind, etc. Team members then take several minutes to write on post-it notes items that push the team towards success, items that drag the team to a stop, items that make the Scrum waters rough. Teams then typically vote for the identified items upon which to focus more in-depth discussion.

Derby and Larsen [2] promote the use of activities within retrospectives as they serve to (i) encourage equal participation, (ii) focus the conversation, and (iii) encourage new perspectives [2]. The authors further recommend that activities should keep participants engaged and that the team should vary the activities that are done within a retrospective to prevent a loss of interest [2]. Derby and Larsen [2] provide both a framework of how to conduct retrospectives as well as a number of retrospective activities. Many more retrospective activities can be found on the internet.

2.2. Games

Suits writes *The Grasshopper: Games, Life, and Utopia* in an attempt to “discover and formulate a definition [of games], and to follow the implications of that discovery even when they lead in surprising, and sometimes disconcerting, directions” [9, p. ix]. *The Grasshopper* proposes, and defends, the following definition for games:

To play a game is to attempt to achieve a specific state of affairs [prelusive goal], using only means permitted by rules [lusory means], where the rules prohibit use of more efficient in favour of less efficient means [constitutive rules], and where the rules are accepted just because they make possible such activity [lusory attitude] - [9, p. 41].

Suits [9] uses golf to defend his definition of games, explicating that to get a golf ball into the hole (prelusive goal) in the simplest way you would pick it up and drop it in the hole, but by doing so you would not actually be playing golf. Only by attempting to hit the ball with a stick (lusory means) and following the intricate set of golf rules (constitutive rules) does the player choose to play golf (lusory attitude).

While gamification research (see [10]) is the current trend in information systems (IS), the use of games is also an established practice. Baker et al. [11] have developed *Problems and Programmers*, a physical card game used to teach software engineering processes in a competitive, face-to-face structure. *Problems and Programmers* focuses on developing collaborative learning and providing immediate feedback to the learner. Baker et al. note that “most of our test subjects felt that playing the game was both a useful lesson and an enjoyable experience” [11, p. 14].

Fernandes and Sousa [12] have developed a competitive, physical card game to help teach Scrum principles in the classroom. *PlayScrum* [12] extends the card game *Problems and Programmers* [11]. Similar to *Problems and Programmers*, *PlayScrum* is a face-to-face game with a focus on visual game play. *PlayScrum* “is simple and fun to play, allows for

collaborative learning, and provides almost immediate feedback to players about the lessons to be learned” [12, p. 59]. *PlayScrum* has been found to be effective in teaching Scrum principles [12].

Paasivaara, Heikkilä, Lassenius, and Toivola [13] explore the use of *The Scrum LEGO Challenge* for teaching Scrum principles, with a focus on (i) Scrum process and rules, (ii) requirements management and customer collaboration, (iii) estimation (iv) working on teams, and (v) visualizing work and progress [13]. The authors have found that the sight of the LEGOs alone makes the game players smile, and that the game was a good resource for teaching Scrum [13].

2.2. Gamification and Game Design

McGonigal [10] notes that game design benefits from the research on positive psychology (see [14]). Positive psychology looks at the productive aspects of people over traditional research into psychological disorders. McGonigal believes that “all of the neurological and physiological systems that underlie happiness – our attention systems, our reward center, our motivation systems, our emotion and memory centers – are fully activated by gameplay” [10, p. 499].

While investigating gamification, McGonigal [15] identifies four elements of game design that promote success: (i) urgent optimism, (ii) social fabric, (iii) blissful productivity, and (iv) epic meaning. Urgent optimism is a feeling that you are close to success, that you have the ability to complete what you are doing. Social fabric represents the bonds that we create, in virtual or real worlds, with others. Blissful productivity reflects the feeling that you enjoy the work you are doing. Finally, epic meaning is the belief that what you are doing is bigger than yourself, bigger than what you can do alone.

3. Research Methodology

To ensure the validity of the game-based retrospective for subsequent research requires a regimented approach to the game design. Similar to previous studies, the game’s initial design is verified through actual observed game play [16] followed by semi-structured questionnaires completed by the game players [11, 12]. The following sections outline the game concept, game design, and the initial validation of the game.

3.1. *Don’t Break the Build*

Don’t Break the Build is a turn-based physical card game designed for play by Scrum teams conducting regular retrospectives. Designed to semi-replicate a typical Sprint, players work to complete 5 daily scrums within the allocated Sprint timebox. During play, each member of the team takes a turn by drawing from a deck of Daily Scrum cards. Daily Scrum cards lead the players to share specific types of observations made during the previous Sprint (e.g., an innovation, something that was learned, a good practice).

Within the Daily Scrum cards are special You Broke the Build cards. When a You Broke the Build card is drawn, the player must draw a card from the Broken Build stack. Broken Build cards direct the team to collaboratively discuss and agree upon a resolution to potential issues (e.g., what is the team’s process for changing the Sprint scope, how does the team encourage collaboration at the daily scrum). Once all team members have taken a turn, the team receives a Daily Scrum Token. The team must collect five Daily Scrum Tokens before the Sprint timebox expires to successfully complete the Sprint. Agreements made during the Broken Build sessions are added to the team rules for the upcoming Sprint.

3.2. Design

Don’t Break the Build is designed by Scrum professionals with years of industry experience leading Scrum teams through retrospectives. Based on lessons learned from industry leaders such as Derby and Larsen [2], countless internet retrospectives, and many self-developed retrospective activities, *Don’t Break the Build* is designed to inform the continuous process improvement that is the cornerstone of Scrum retrospectives while also serving as a game for Scrum teams to enjoy playing.

Instead of being designed as a competitive game such as *Problems and Programmers* [11] where players are pitted against one another, *Don’t Break the Build* is a competitive game where players work together as a team to beat the clock. Additionally, although not a traditional learning game like *Problems and Programmers* [11], there are structured learning aspects of *Don’t Break the Build* encountered as the team identifies and resolves issues affecting their real Scrum team.

The design for *Don’t Break the Build* focuses first on meeting the criteria established for defining a game [9] in that it is structured to include (i) prelusory goals, (ii) lusory means, (iii) constitutive rules, (iv) lusory attitude, and (v) lusory goal.

Prelusory Goal - The “specific state of affairs” [9, p. 41] that is the goal of a retrospective is inspection and adaption of the team and team

processes. The goal of a game-based retrospective, should, therefore, achieve this desired effect. The prelusory goal of *Don't Break the Build* is to aid the team in this process.

Lusory Means - The lusory means of *Don't Break the Build* are the game rules that define everything from the number and types of cards used to the manner in which the game is played. The rules direct the players to draw one card at a time and to proceed in a clockwise manner. Further, the rules direct the players as to what must be completed in response to each card drawn.

Constitutive Rules – The framework developed by Derby and Larsen [2] for conducting a retrospective provides guidance that helps teams identify, prioritize, and adapt processes based on team input in an efficient, directed manner. The

safety, and (vii) entertainment. *Don't Break the Build* focuses on the game design elements identified by McGonigal [18]:

Epic Meaning – The team must work together to successfully complete the Sprint. Only through teamwork and collaboration can they succeed.

Urgent Optimism – The clock is working against the team as they play, but by working together they can complete the Sprint in time.

Social Fabric – The team interacts and collaborates, quickly working to make decisions that are agreeable to all.

Blissful Productivity – The team chooses to work through issues and decision making in order to win the game, activities that they might in normal circumstances do begrudgingly.

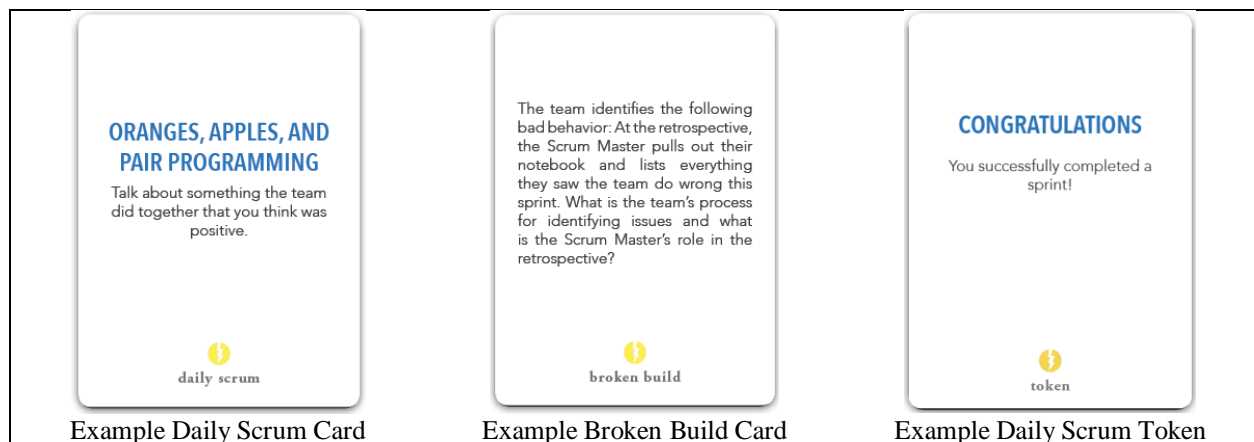


Figure 1 - Prototype game items

constitutive rules that define *Don't Break the Build* force discussion of specific topic areas, randomize the order of topics to be discussed, and force teams to limit discussion in order to beat the clock.

Lusory Attitude – Team members must be willing to play together and work towards process improvement by following the game's lusory means and constitutive rules enabling a lusory attitude.

Lusory Goal - The team must understand how the game is won or lost. *Don't Break the Build's* lusory goal is to complete the Sprint within the allotted time-box. Failure to complete the Sprint constitutes losing the game.

While Suits [9] provides guidance for what constitutes a game, other researchers provide varying levels of information on creating an engaging game design. Baker et al. [11] note that fun and ease of play are key components of their game design while Ho et al. [15] note seven game design elements: (i) goal, (ii) rules, (iii) competition, (iv) challenge, (v) fantasy, (vi)

3.3. Preliminary Game Testing

To test the playability and enjoyment of *Don't Break the Build*, the game was subjected to three rounds of observed game play using four teams. Initial testing was conducted using two established Scrum teams. After the results of the initial testing were analyzed, minor changes were made to the rules and the game was retested with a third established Scrum team. Following this testing, the game was tested a fourth time with a newly formed Scrum team to evaluate differences in playability and enjoyment based on team experience.

3.4. Test 1

The goal of Test 1 was to validate the basic game design and mechanics and to ensure the game is enjoyable to play. Test 1 was used to verify that the

rules and directions are easy to follow and enable the teams to play the game.

Test 1 consisted of observing two different Scrum teams playing *Don't Break the Build*. Both teams agreed to observed game play during their regularly scheduled retrospective. The game designers had served as Scrum Masters for both teams, and both teams were accustomed to activity-based retrospectives.

Team 1 consisted of five members of the Scrum team, however neither the Product Owner nor the Scrum Master was present for the retrospective. Team 2 consisted of six members of the Scrum team including the Product Owner and an acting Scrum Master. The game and rules were provided to the teams, and a game designer was present to respond to questions arising during game play. Both Team 1 and

players provided over forty constructive thoughts and recommendations that supported their evaluation of the game.

Of the comments submitted, the most prominent issues identified, positive and negative, were related to: the instructions on the cards (15 issues), the physical cards (12 issues), and the time limit (12 issues). The following paragraphs discuss the comments from these groups.

The Daily Scrum cards in *Don't Break the Build* guide discussion. For example, "Talk about something you think inhibits team success". The card does not specify whether this should come from the most recent Sprint or if something in the past is acceptable. Further, the rules of the game state "The player, and the team if necessary, works to complete the task on the Daily Scrum Card". The language on the card, and

Table 1 - Questionnaire results

Questions	Team 1	Team 2	Team 3	Team 4	Average
How enjoyable is <i>Don't Break the Build</i> to play? (1 – not enjoyable at all, 5 – very enjoyable)	4.2	4.3	4.0	4.5	4.3
How difficult/easy is <i>Don't Break the Build</i> to play? (1 – not easy at all, 5 – extremely easy)	4.2	4.5	3.8	4.3	4.2
How well does <i>Don't Break the Build</i> help to identify issues (positive or negative) within the team? (1 – not well at all, 5 – extremely well)	3.4	3.5	4.0	4.3	3.7
How well does <i>Don't Break the Build</i> serve as a retrospective activity? (1 – not well at all, 5 – extremely well)	3.8	3.8	4.6	4.8	4.2
How often would you like to play <i>Don't Break the Build</i> ? (1 – never again, 5 – every retrospective)	3.8	3.3	3.4	3.8	3.6
Average	3.8	3.9	4.0	4.3	4.0

2 collected all five Scrum Tokens and "won" the game. After the game, team members completed a questionnaire (see Table 1). The questionnaire was based on Baker et al. [11] whose questionnaire supported the development of the *Problems and Programmers* game described above.

3.4.1. Results. Table 1 shows the quantitative results of the questionnaire completed by participants after both observed play sessions. The quantitative results show that, overall, the teams found the game to be enjoyable and easy to play but had concerns with how well it helped to identify issues and serve as a retrospective activity.

The questionnaire also encouraged participants to provide additional comments to help identify improvements to the game by stating: Please provide any comments/suggestions/recommendations that would help improve *Don't Break the Build*. The

supporting rules, are intentionally left vague to allow the team freedom to address issues that are important to the team without feeling constrained by the game.

There are a number of suggested improvement for the physical cards themselves. Players suggest integrating the Broken Build cards directly into the playing deck and eliminating the Daily Scrum – You Broke the Build cards. This change requires modifications to the card back colors.. The comments do suggest changes that should be implemented before final adoption of the game in order to improve the ease of the game play.

The time limit imposed on the game serves as the lusory goal that determines if the team wins or loses the game. The time limit is also the item identified the most in the comments section of the questionnaire. While exploring other potential lusory goals during game design, most were eliminated as they invoked a player versus player aspect to the game whereas the

game is intended to help foster team building, communication, and collaboration. Further, the time limit was specifically chosen as the lusive goal as it reinforces the timebox nature of Scrum.

The Scrum Guide instructs that “All events are timeboxed events, such that every event has a maximum duration” [1, p. 9]. With the exception of Sprints, all Scrum ceremonies “may end whenever the purpose of the event is achieved, ensuring an appropriate amount of time is spent without allowing waste in the process” [1, p. 9]. Ruben notes that timeboxes are “a time-management technique that helps organize the performance of work and manage scope” [19, p. 62]. Ruben [19] also states the benefits of timeboxing include (i) forcing prioritization, (ii) demonstrating progress, (iii) avoiding unnecessary work, (iv) motivating closure, and (v) improving predictability.

However, the players’ responses cannot be ignored. A potential solution to this problem is to provide better explanation of how the game is intended to be used within a retrospective. An unintended consequence of the game, as one player noted, was that “With no artifacts (like stickies), it is very hard to remember what everyone said and take action on them”. While the game designers intended for the team to record decisions made by the team during play, this was not stated clearly in the rules. Cards were designed to help the team develop process (e.g., “The customer has an urgent need and wants the team to work on it immediately. What is the team’s process for adding/removing items from a Sprint?”), but directions were not included that instructed the team how to use the game during a retrospective, only how to play the game.

3.4.2. Game Modifications. While many of the recommendations for changes to the card instructions and physical cards are valid, they cannot be implemented during the test process due to time constraints.

To mitigate the time concern, the rules and directions have been modified to direct the Scrum Master to facilitate the game and to record decisions and action items generated by the team during game play. This provides the artifacts that were missing from the game play during Test 1 and helps the team to focus on completing the card tasks.

3.5. Test 2

Following the modifications to the game as outlined above, another observed game play was conducted with a third established team. Team 3 does

not regularly work with the game designers but agreed to allow them access to the retrospective to observe the game play. Team 3 consisted of nine members of the Scrum team, including their regular Scrum Master and Product Owner. One member of the team participated in the retrospective by phone as they were traveling on the day of the retrospective. The team collected four Scrum Tokens and therefore “lost” the game.

3.5.1. Results. Table 1 shows the quantitative results of the questionnaire completed by participants of Team 3 playing the game with the modified rules. The quantitative results show that the team found the game to be enjoyable and easy to play and also found it helped to identify issues and serve as a retrospective activity.

Team 3 submitted 30 comments which included 9 on the physical cards, 8 on the card instructions, and 5 on the time limit. While these comments mirror the comments made by the first two teams, the quantitative results show improvement to the games usage as a retrospective.

3.5.2. Game Modifications. No modifications to the game were made prior to the third, and final round, of observed game play.

3.6. Test 3

Test 3 involved a newly formed Scrum team completing its first Sprint. The game was played during the first retrospective, and only included four team members, including the Product Owner. Due to its size, the team does not yet have a full time Scrum Master. Since the team is new, there is limited shared history among team members, they have not experienced issues as a group and learned how to deal with them. The team was unable to complete the game and only collected two Daily Scrum Tokens.

3.6.1. Results. Table 1 shows the quantitative results of the questionnaire completed by participants of Team 4 playing the game with the modified rules. Team 4 had the highest overall average (4.3) when looking across all factors. However, the team only submitted five comments.

4. Conclusions and Future Work

The observed game play with four Scrum teams validates the overall game mechanics and concepts and provides valuable feedback on areas of the game that needed improvement. *Don’t Break the Build*

successfully meets the game criteria established by Suits [9] and leverages game design elements identified by McGonigal [18] as supported by the quantitative questionnaire results showing that the game is enjoyable and easy to play. While some improvements were made during the test session, further improvements still need to be made.

4.1. Improvements

The observed game play highlighted many

to the cards which identifies precisely which cards to use based on the number of players.

This solution also has the benefit of solving another player identified issue regarding duplicate cards. Several team members noted:

- “A couple more variations in cards/questions to facilitate more discussion and identify more problems/issues”
- “Reduce the number of duplicate cards”
- “We had a lot of repeat cards and answers”

Currently, as the players randomly choose the Daily

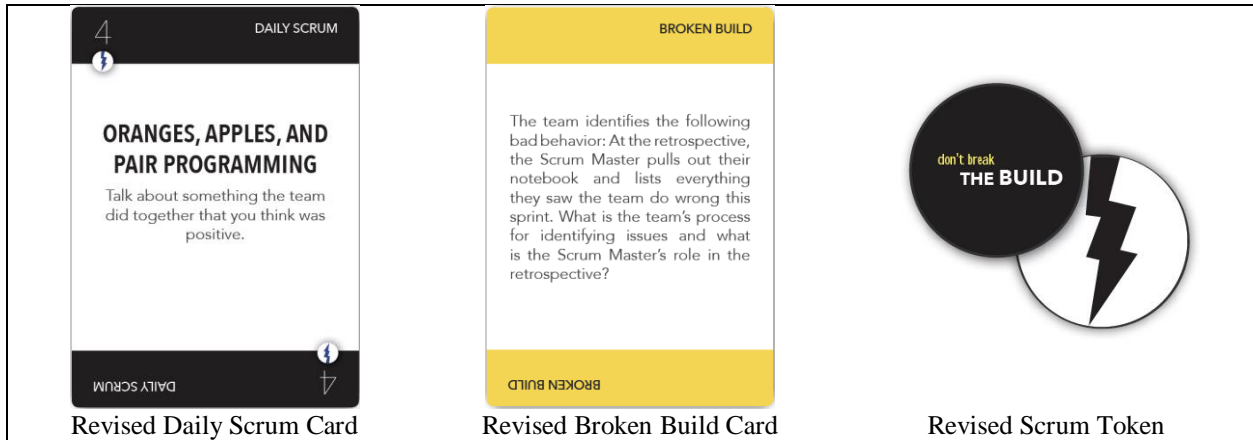


Figure 2 - Revised game items

modifications that can be made to game mechanics and questions in addition to the further review of the game’s time limit.

4.1.1. Physical cards. It is immediately recognized that the different decks of cards for the game are an unnecessary complication. In the current game structure, You Broke the Build cards are used in the game deck to signal the player to draw from the Broken Build deck. One respondents noted that “The separation of Broken Build cards from the rest of the daily scrum deck seems pointless. These cards should be shuffled into the same deck as the other question cards with the same card back”.

While this solution is relatively simple, it was not done during the test session as the Daily Scrum Cards and Broken Build cards had different colored card backs and would, therefore, be easily recognized before being drawn.

The need to sort and count cards to set-up for game play is another hinderance to an enjoyable game. A respondent suggested “Have I Broke the Build cards, and Broken Build cards more distinguishable when flipping through”. This problem can be remedied through a modification to the cards by adding an icon

Scrum cards for the playing deck, there is a chance that duplicate cards will be selected, and unique cards will be left unseen. By pre-defining which cards are used based on the number of players, duplicate cards will only come into play with larger teams.

4.1.2. Card instructions. Negative cards, unknown time period, and knowing who can respond are the three main issues with the card instructions. Regarding negative cards, a team member noted “Some of the questions . . . tend to focus more on negative attributes of the team and Sprint.”. Despite the team’s feeling that there are more negative cards than positive, there are not. For each type of card, there is a negative, a positive, and a neutral card. However, based on the discussion above regarding the physical cards, there is a possibility that due to the random draw, teams were experiencing a higher number of negative cards. The solution as stated above, to pre-define specific cards based on team size, will also help to rectify this problem.

Additionally, it is important to note, as one team member did, that talking about uncomfortable issues is important during retrospectives: “I really felt like it forced our team to talk about "negative" topics that we either don’t think about regularly, or we avoid talking

about because it's uncomfortable". Often, real process and team improvement stems from these uncomfortable discussions.

The unknown time period is a problem that periodically plagues Scrum teams during regular retrospectives. Respondents commented:

- "Should these be about the past Sprint, or is any time period okay."
- "Sometimes the team wasn't certain whether the answers should come out of this current Sprint or overall, not a big deal, but an observation."
- "It was unclear if you should talk about this Sprint or overall. This may be ok."

There are issues that affect the team from Sprint to Sprint, and then there are issues that have happened in the past that team members want to discuss. During regular retrospectives, the Scrum Master will help the team navigate these waters to determine what is, and is not, in scope for the retrospective. Resolution of this issue may include more clarification within the game rules to help the team prepare for this question during play. The team should, while setting up for the game, decide whether they want to focus solely on the current Sprint or if they want to leave discussion open for issues from previous Sprints.

The issue of who should respond to a card is similar in nature to the time period issue. Respondents noted that "Rules about who can talk when a card is drawn aren't clear enough" and "We were not sure when the whole team was supposed to discuss or if a single person was meant to answer". During play, these types of issues manifest in two ways. First, a player may draw a card that they legitimately do not have a response to. Secondly, a player who is not drawing may resonate passionately with the topic drawn by another player. In either case, from a retrospective perspective, the game should handle either situation.

The simplest solution, again, is to expand the rules to identify that either of these cases may occur during game play, and that the team should decide how they want to handle each case. A player with no specific response may choose to ask another player for help while a player who wants to respond to another player's card may be allowed to do so. Caution must be taken here, however, as there are potential negative aspects to both of these situations. First, players may use the "ask a teammate" rule in order to shy away from responding to questions that they do not want to answer. This would reduce the ability of the game to force hard conversations that might otherwise not be had.

Allowing non-drawers to respond to a card might allow dominant team members to control the game and

what is being discussed. A benefit of activity-based retrospectives and of this game-based retrospective is that by design they strive to limit the ability of any team member to have an overbearing impact on the discussion. More research and discussion on this topic is necessary before implementing changes.

4.1.3. Time limit. The changes during observed game play directing the Scrum Master to record observations and action items resulting from discussion during the game have already improved the game's usability during retrospectives. However, the issue of game's time limit remains. As stated previously, the time limit is a design decision that reinforces the timebox nature of Scrum and represents the tradeoffs that must be made during a Sprint to ensure all items are completed. It also helps to keep conversations on topic. Finally, it represents the win/loss aspect of the game. Failure to apply a lusive goal to the game precludes the game from being a game.

Discussion with other game designers have yet to yield viable options to the time limit. While there are a number of cooperative card games, many of these still employ a competitive edge. As *Don't Break the Build* is geared, in part, to build team collaboration and cooperation, it is imperative that competition between team members within the game is avoided. This remains the largest question to solve prior to release of the game.

4.2. Future Work

The results of the four observed game play sessions show that *Don't Break the Build* is an enjoyable, easy to play game that has potential for use as a Scrum retrospective tool. Improvements, as identified above, are necessary to increase the ease of play and other aspects of the game, but the basic game mechanics and topics are in place.

For the game, the next steps are to make the discussed changes and retest the game to validate the correctness of those changes. This should be done with a larger and more diverse set of Scrum teams. Results will be measured using the existing questionnaire, although observation of the game play may not be practical. After validation of the changes, *Don't Break the Build* will be positioned to support further research, including the testing of Scrum retrospective effectiveness measures [4].

The research is also intended to support the deliberate and scientific development of games to support improvements within software development and the wider arena of information systems and to

support the research showing the positive effects of games on team development.

Two unexpected research questions were identified during the observed game play. First, is it a good idea to introduce success and failure (winning or losing the game) into a retrospective? As retrospectives are aimed to inspect and adapt process and team performance, will the possibility of failing change the retrospective dynamic?

Second, is there an unintended effect on the role of the Scrum Master by using a game such as *Don't Break the Build* during a retrospective? Scrum Masters are responsible for facilitating the Scrum retrospective, and for determining the activities and guiding discussion during the retrospective to help the team explore and uncover issues. While *Don't Break the Build* could be considered a good training aid for new Scrum Masters or a good activity for when a Scrum Master is absent, there is also the potential that Scrum Masters might over-use the game instead of taking the time to develop team-specific retrospectives, thus potentially limiting their insight and value to the team.

5. References

- [1] K. Schwaber and J. Sutherland, "The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game," Scrum.org, 2017.
- [2] E. Derby and D. Larsen, *Agile Retrospectives: Making Good Teams Great*, Dallas: The Pragmatic Bookshelf, 2006.
- [3] A. Przybylek and D. Kotecka, "Making Agile Retrospectives More Awesome," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, 2017.
- [4] D. Marshburn, "Scrum Retrospectives: Measuring and Improving Effectiveness," in *Proceedings of the Southern Association for Information Systems Conference*, Atlanta, 2018.
- [5] K. Beck, M. Beedle, A. van Bannekum, A. Cockburn, W. Cunningham, M. Fowler, J. Greening, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas, "Principles Behind the Agile Manifesto," 2001. [Online]. Available: <http://agilemanifesto.org/principles.html>. [Accessed 02 06 2018].
- [6] K. Kumar, "Post Implementation Evaluation of Computer-Based Information Systems: Current Practices," *Communications of the ACM*, vol. 33, no. 2, pp. 203-212, 1990.
- [7] O. Salo and P. Abrahamsson, "An Iterative Improvement Process for Agile Software Development," *Software Process Improvement and Practice*, pp. 81-100, 2007.
- [8] B. Collier, T. DeMarco and P. Fearey, "A Defined Process for Project Post Mortem Review," *IEEE Software*, vol. 13, no. 4, pp. 65-72, 1996.
- [9] L. Gonçalves, "Run the Sailboat Agile Exercise or Sailboat Retrospective," 25 05 2018. [Online]. Available: <https://luis-goncalves.com/sailboat-exercise-sailboat-retrospective/>. [Accessed 02 06 2018].
- [10] B. Suits, *The Grasshopper: Games, Life, and Utopia*, Toronto: University of Toronto Press, 1978.
- [11] J. McGonigal, *Reality is Broken: Why Games Make Us Better and How They Can Change the World*, Penguin Press, 2011.
- [12] A. Baker, E. Navarro and A. van der Hoek, "An Experimental Card Game for Teaching Software Engineering Processes," *The Journal of Software and Systems*, pp. 3-16, 2005.
- [13] J. Fernandes and S. Sousa, "PlayScrum - A Card Game to Learn the Scrum Agile Method," in *International Conference on Games and Virtual Worlds for Serious Applications*, Braga, Portugal, 201.
- [14] M. Paasivaara, V. Heikkilä, C. Lassenius and T. Toivola, "Teaching Students Scrum Using LEGO Blocks," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014.
- [15] M. Seligman and M. Csikszentmihalyi, "Positive Psychology," *American Psychologist*, vol. 55, no. 1, pp. 5-14, 2000.
- [16] J. McGonigal, "How a Game Saved My Life," [Online]. Available: <http://janemcgonigal.com/videos/>. [Accessed 02 06 2018].
- [17] K. M. Gerling, J. Schild and M. Masuch, "Exergame Design for Elderly Users: The Case Study of SilverBalance," in *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*, 2010.
- [18] P. Ho, S. Chung and M. Tsai, "A Case Study of Game Design for E-Learning," in *International Conference on Technologies for E-Learning and Digital Entertainment*, 2006.
- [19] J. McGonigal, "Gaming Can Make a Better World," [Online]. Available: <http://janemcgonigal.com/videos/>. [Accessed 02 06 2018].
- [20] K. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Addison-Wesley, 2013.