Summary

From our prespecialization research and previous research, we found that facilitating Sprint Retrospectives (SRs) can be a difficult task. People lacked motivation for participating in SRs, which produced multiple underlining issues. In this paper, we look into the issue of participants not participating in submitting feedback to a SR. More specifically we try to improve the issue by introducing nudging into the facilitation of a SR. We developed a SR tool for developer teams which consists of features that allow the teams to facilitate a full SR meeting.

For the developed platform, we decided to use four different nudges to see if we could affect the participants to submit more and better feedback. More specifically, we designed the nudges for throttling mindless activities, creating friction, instigating empathy, and raising the visibility of user actions, to nudge the participants towards adding more feedback. The instigating empathy- and raising the visibility of user actions nudges were combined, meaning whenever a participant submitted a piece of feedback, a counter would increment and a smiley would transition to become happier until the threshold specified by the counter was reached. The throttling mindless activities nudge was designed as a pop-up that would appear to the participants when joining a SR if they did not submit enough feedback during the feedback collection phase. The creating friction nudge was designed as a color indicator with the purpose to nudge the participants towards submitting better feedback. The color indicator would appear around the feedback input box, so whenever the participants would begin writing their feedback, the color indicator would change color from red towards green based on the semantic value of their current input.

To test whether the four nudges could increase the quantity and quality of the submitted feedback in a SR, we designed and conducted a 2-condition within-subject experiment. The participants consisted of 2 computer science study groups, with 4 participants in each group. In the experiment, we exposed the participants, through the platform, to both no nudging and nudging. The study had a duration of 8 weeks (4 SRs in total), where the first two SRs would have nudges disabled, and the last two would have nudges activated. The participants were aware of whether nudging was disabled or activated. After the study of 8 weeks, we also conducted semi-structured interviews with each of the participants individually. The goal of the semi-structured interview was to see how the participants behaved towards the feedback quantity and quality nudges.

From the quantitative data collection, we found an increase in the submitted amount of feedback, more specifically it increased by 37% when the participants were exposed to the nudges. We conducted a paired samples t-test for these results, which indicated that the difference in data was not significant. For the quality of the submitted feedback, we found that the semantic value increased by 40% when nudging was activated. Again, we conducted a paired samples t-test and found that the difference in data was not significant.

From the qualitative data collection, we asked the participants questions regarding the nudges in a semi-structured interview. For the three quantitative nudges, we found that the raising the visibility of user actions and instigating empathy nudges had the largest impact on the participants as they motivated them to submit more feedback. The two nudges also turned their SRs into a more game-based experience compared to before the nudges were activated. For the qualitative nudge, i.e the creating friction nudge, we found that the participants had a mix of experiences with the nudge. For some participants it made them reflect more on their feedback, while others did not think about it, but discovered that the team spent less time discussing the meaning of the submitted feedback due to the increased detail of the submitted feedback.

The study had multiple limitations which might have affected the validity of the paper. We had a limited duration of time for conducting the study and we only had two study groups as participants. However, from the feedback collected from the interviews as well as the results, it could indicate that the nudges had an overall positive effect. Even though the nudges had a positive effect on the participants, we also learned that the nudges could be a source of frustration. Conducting pilot tests on the implementation of the nudges could have improved the experience of the nudges for the participants. We also did not include a measurement that could explain the effect of each implemented nudge, this could have been interesting to include to elicit the nudges which had the most impact.

Since our findings display an opportunity of using nudging in the work environment, we do encourage other researchers to extend this exploration. The paper only investigates the nudges throttling mindless activities, creating friction, instigating empathy, and raising the visibility of user actions, therefore an extension of this exploration could include other nudges in different categories and perhaps different combinations of nudges.

Improving Developer Feedback Quality and Quantity in Sprint Retrospectives using Nudging

Christopher K. Sørensen
Aalborg University
Aalborg, Denmark
cksa17@student.aau.dk

Frederik B. Jacobsen
Aalborg University
Aalborg, Denmark
fjacob17@student.aau.dk

Nicolai S. Bjerring
Aalborg University
Aalborg, Denmark
nsbj17@student.aau.dk

Abstract

Nudging software developers to provide more, and better, feedback in Sprint Retrospectives can produce more effective Sprint Retrospectives, but developers tend to lack motivation for these meetings. We conducted a mixed-method study testing the quantity and quality of feedback with and without nudging for 8 weeks. Throughout the 8 weeks, the participants conducted two Sprint Retrospectives without nudging, and two Sprint Retrospectives with nudging activated. The results showed a 37% increase in the amount of submitted feedback and an increase in average semantic value of 40%. Participant interviews revealed an increase in motivation when exposed to nudging by the majority and overall satisfaction with the platform.

CCS Concepts: • **Human-centered computing** → *Collaborative interaction*; • **Software and its engineering** → *Agile software development*.

Keywords: nudging, human-computer interaction, sprint retrospective

1 Introduction

According to the 15th annual *State of Agile* report, agile software process models increased in popularity by 49% between 2020 and 2021 [4, 5]. Sprint Retrospectives (SRs) rank second-highest of activities incorporated in agile software teams, only beaten by the daily stand [4, 15]. In previous prespecialization research [2], we extended existing research, by Matthies et al. [9], on issues with SRs and their associated value in software development teams, underlining issues with multiple themes, including *data collection*, *motivation*, and *action*. Interviews and observations made during the prespecialization research displayed a lack of software developers engaging in SRs, even though the majority of the participants believed that SRs were useful. More specifically, upwards of 50% of the participants observed did not provide feedback during SRs. This paper investigates whether the introduction of *nudging* into SRs can mitigate this issue.

Our motivation for nudging is inspired by *Thaler and Sunstein* who state that nudges can be used to motivate people in different topics and scenarios ranging from lowering a household's energy consumption to improving your health. Nudges are seen in both digital and physical environments but can be described as "*any aspect of the choice architecture that alters people's behavior in a predictable way without forbidding any options or significantly changing their economic incentives*" [13].

In this study, we alter the choice architecture when facilitating SRs by integrating nudges in the process of contributing feedback to a SR. From this, we looked into whether such nudges could assist the user's thought process by nudging them to reflect more on their feedback [13]. We implemented a tool that integrated a total of four nudges, where each nudge had a singular focus, either throttling mindless activities, creating friction, instigating empathy, or raising the visibility of user actions. The tool was developed to allow teams to facilitate SRs where the users were primarily exposed to the nudges when submitting feedback to SRs.

Through a period of 8 weeks, 2 teams used our tool to facilitate 8 SRs in total. In the first half of the facilitated SRs, the tool did not display any nudges, and in the second half, the nudges were activated, exposing the four nudges to the users. Only exposing the teams to the nudges for the last part of the experiment allowed us to compare whether the nudges affected the quantity and quality of the submitted feedback.

The study revealed an increase in the quantity and quality of submitted feedback when nudging was activated, where almost all users increased the amount of feedback submitted between SRs and the semantic value of the submitted feedback. The users also shifted to a more evenly distributed amount of feedback pr. user. The majority of the participants felt that the platform increased the structure of their SRs, and streamlined the facilitation of SRs. In general most of the participants found the nudges fun, describing it "gamified" their SRs to some extent.

, ,

This paper makes two contributions to the field of HCI research:

- We build on the existing systematic categorization of nudges presented by Caraban et al. [5] and evaluate the implementation of 4 nudges; throttling mindless activities, creating friction, instigating empathy, and raising the visibility of user actions.
- We conducted a study for +8 weeks, where 2 study groups interacted with the implemented nudges. In the study, we explored whether such nudges could be used to alter the users' behavior to increase the quality and quantity of their submitted feedback.

This work shows how nudging can be used to influence participants in SRs to increase the quality and quantity of their submitted feedback.

2 Related Work

The HCI community does not contain much research regarding SRs, yet there exist interesting papers discussing problems within SRs with propositions on how to deal with them. We investigated whether nudging has ever been proposed, and did not find any related work. In this section, we discuss issues with SRs and how extending previous research within nudging could help mitigate these.

Sprint Retrospectives

Running SRs is not an easy process, which Przybyłek and Kotecka [11] try to accommodate by highlighting that facilitating SRs can be challenging since a prescription for facilitating a SR does not exist. Furthermore, they also talked with a focus group consisting of people from the software industry with varying expertise. From the focus group, they discovered that the SR was the least favorite meeting, where some believed it added no value to the team and others had simply never tried it in practice but only knew it theoretically. Based on these findings they introduced collaborative games for facilitating SRs. They came up with over 100 different games based on previous literature, where four games were hand-picked based on problems found in other literature and the focus group. These four games were tested on three different development teams and the results were positive. The teams continued to use the introduced games after the experiment, however, one game cannot suit all teams, which is why the scrum master or facilitator needs to choose the correct game based on the problems their team encounters.

Matthies et al. [9] believed that even though these game-based SRs solve some of the issues found in SRs, they still lack explicit instructions on which game to use based on specific SR problems. In their paper, they extend the work on collaborative games for SRs and try to map frequently found problems to specific games. Their approach for identifying frequent and common problems in SRs was to search through others' work but also from popular Scrum practitioner websites. These were the most common problems found in SRs:

- All Talk No Action
- Too Repetitive
- No Preparation
- Blame Game
- Not Speaking Up
- Taking It Personally
- Group Think
- Focus on Negatives
- Complain Game

Afterward, they mapped these problems to already existing activities, explaining which activities solved which problems. To test their mapping, they ran an experiment on multiple software teams using the found activities, where they observed their SRs and conducted interviews with the teams. Overall, they found that some of the mappings were true and that the already existing activities solved some of the problems without explicitly saying so. Therefore, they could create a more detailed description of which activities to use to alleviate specific SR problems.

It is clear from the aforementioned studies that facilitating SRs is challenging and does propose multiple issues. Different solutions have been proposed which involve turning SRs into game-based events, where the activities are aimed at specific issues that appear in the team, either in their internal processes or in their facilitation of SRs. In this paper, we narrow down the focus and look into the initial part of SRs, which is the feedback phase.

Nudging

In 2008 Thaler and Sunstein introduced *nudging* [13] as a way of changing people's behavior in predictable ways by using systematical choices in the design architecture. As described by Caraban et al. [3], this was *"eagerly adopted in HCI"*. Thaler and Sunstein describe a nudge as [13]:

> *Any aspect of the choice architecture that alters people's behavior in a predictable way without forbidding any options or significantly changing their economic incentives. To count as a mere nudge, the intervention must be easy and cheap to avoid. Nudges are not mandates. Putting fruit at eye level counts as a nudge. Banning junk food does not.*

In a paper from 2018 [12], Renaud and Zimmerman investigate how nudges can influence the complexity of user passwords. They performed a series of studies to evaluate the effects of different nudges. They defined two distinct types of nudges, i.e. *simple-* and *hybrid nudges.* Simple nudges are basic visual nudges that aim to guide the user's choice without being too complex to the extent of requiring reflection, whereas hybrid nudges are compositions of nudges that are more complex and encourage users to reflect on their input. Their studies show no significant difference between the produced password complexity of a group with no nudging enabled and a group with simple nudging enabled. They did, however, find a significant difference in produced password complexity with a group that was exposed to hybrid nudging. They describe the findings as being a result of the hybrid-nudge group clearly understanding the benefits of a strong password, as well as the risks of a weak password. In this study, the authors implemented an incentive, where users' passwords expire after a certain period; the stronger the password, the longer the expiration time. Renaud and Zimmerman [12] briefly touched on the topic of types of nudges, as they mention *incentive* and *reminder* as categories. In a paper from 2019 [3], Caraban et al. present a framework for technology-mediated nudging, which clarifies the types of nudging as well as when to use them.

Caraban et al. present 23 different nudges within 6 categories and map the nudges into Fogg's behavioral model [6]. In this paper, we utilize the research by Caraban et al. to determine which nudges to include in our study, as we extend nudging research within HCI by applying nudges to a new context; SRs.

3 Designing the *retros* platform

In recent research, 10 years after Thaler and Sunstein presented nudging, Caraban et al. [3] presented the HCI design space for nudging, with 23 nudging techniques mapped into Fogg's behavior model [6] as either *signal*-, *spark*-, or *facilitator* triggers. In this paper we disregard facilitator triggers, which aim to simplify tasks or make certain behavior easier, complementing users who may not have the ability to do a task. The task of providing feedback, i.e. giving your opinion, to SRs is already simple, and therefore we assume no facilitator trigger would have a significant impact on the study. Instead, we focus on the signal- and spark triggers, which mainly focus on increasing motivation or the user's perception of gain from doing the task. Signal triggers can be useful in situations with discrepancies between a user's intentions and actions, e.g. by creating friction in terms of doubt, discomfort, etc. Spark triggers can be used to increase motivation by leveraging social comparison, perception of the possibility of a loss, etc. In this paper, we introduce 4 nudges; 2 mapped to the spark trigger and 2 mapped to the signal trigger, which will be described in further detail below. The study is limited to 4 nudges to limit the complexity of the implementation.

Caraban et al. also presented 6 categories of nudges and their use-cases as [3, appx. C]

1. *Facilitating nudges*: aim to reduce the mental or physical effort for a task.
2. *Confronting nudges*: aim to break mindless behavior and spark a user reflection.
3. *Deceiving nudges*: aim to change the perception of certain outcomes to promote these.
4. *Social influencing nudges*: aim to underline what is expected of people in a context and promote this.
5. *Reinforcing nudges*: aim to reinforce certain behaviors by increasing their presence.
6. *Fear nudges*: aim to invoke a feeling of fear or loss by (not) doing a certain activity.

Nudges from the category of facilitating nudges (1) are excluded, as these are all mapped to the facilitator trigger. We expect that fear nudges (6) will not have a significant impact on the study, as the domain of SRs does not come with a possibility of financial, physical, or major psychological risk. Within the category of deception (3), we find nudges such as *placebos*, *biasing the memory of past experiences*, etc. When brainstorming ideas of how to implement the 23 nudges within the domain of SRs, we did not see a potential for any of these nudges. In the next section, we discuss which nudges from the categories of confronting- (2), social influencing- (4), and reinforcing nudges (5) will be included in the study.

Nudges

In the context of SRs, we look further into confronting-, reinforcing- and social influencing nudges mapped to either a spark- or signal trigger. Caraban et al. place no nudges within the contexts of *social influencing signal-* and *reinforcing spark* nudges, and therefore these are omitted from this section.

Confronting spark nudges

Within confronting spark nudges lies

1. *Provide multiple viewpoints*: e.g. provide the user with a second option.
2. *Reminding of consequences*: e.g. underline the risk of not providing feedback.
3. *Throttling mindless activities*: e.g. prompt users to provide feedback before joining a retro if they have not already.

Option 1 can be scaled to fit a certain complexity, i.e. the platform could suggest other words when providing feedback, which could be a product of AI technology integrated into the platform or simply predetermined words or phrases. Previously, fear nudges were dropped, as the domain does not introduce significant risks, and therefore option 2 is not considered. Option 3 has previously been implemented within the domain of Facebook posts, where Wang et al. [14] found that several users changed behavior. We proceed to implement a variation of option 3, which has been proven to be effective in the past.

Confronting signal nudges

Within confronting signal nudges Caraban et al. only place 1 nudge, which is

1. *Creating friction*: e.g. adding a color indicating aureole around the feedback input box, which gradually turns from red to green the better the inputted feedback.

Comparing this option, using a simple implementation of either counting words or semantically addressing the value of a certain phrase with natural language processing, to option 1 from *confronting spark nudges*, i.e. providing multiple viewpoints, this option is more attractive from our point of view, as the nudge kicks in even if the user does not input any feedback. This nudge draws attention to the users' actions [3, appx. C], and it can force attention specifically to the quality of the inputted feedback.

Social influencing spark nudges

Looking further into social influencing nudges, we find

1. *Public commitment*: e.g. transparently notifying the Scrum Master of how much feedback is given pr. developer.
2. *Reciprocity*: e.g. giving the developers something in return before asking them to provide feedback.
3. *Enabling comparison*: e.g. displaying the amount of feedback provided by the most active user to developers.
4. *Raising the visibility of user actions*: e.g. displaying how much feedback the developer has given at the given time.

Options 1 and 2 come with certain drawbacks, e.g. the complexity of determining what items to reciprocate with. Option 3 and 4 are similar nudges, as they both aim to introduce visibility of the task. The difference is between the visibility of a top performer and the user. Option 4 seems most appropriate, as option 3 indicates a competing culture, which is not the point of SRs.

Reinforcing signal nudges

Finally, within reinforcing signal nudges lies

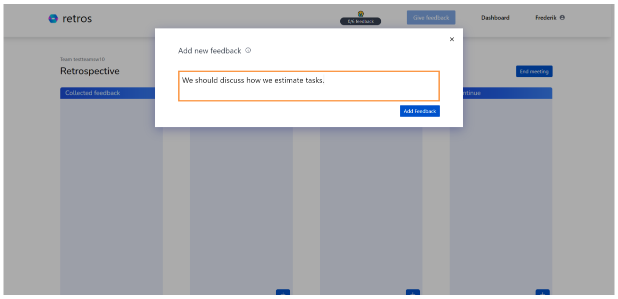
1. *Just-in-time prompts*: e.g. a push notification to remind the user to submit feedback when a time limit has been reached.
2. *Instigating empathy*: e.g. displaying a smiley face, which gets happier the more feedback the user inputs to the system.
3. *Ambient feedback*: e.g. adding lights to the floor to display the shortest route. We found this nudge difficult to exemplify for SRs.
4. *Subliminal priming*: e.g. flashing goal-related words to the user.

During the brainstorming session, we found no obvious implementations of subliminal priming- and ambient feedback nudges, and after researching possible implementations we did not find these nudges suitable for the domain. Instead, we focused on options 1 and 2. Investigating implementations of just-in-time prompts, we learned that these may overlap with nudges for throttling mindless activities within our domain. In conclusion, we decided to implement the nudges presented in Table 1 in our study.

	Spark	**Signal**
Confronting	Throttling mindless activities	Creating friction
Reinforcing		Instigating empathy
Social influencing	Raising the visibility of user actions	

Table 1. The nudges to be implemented.

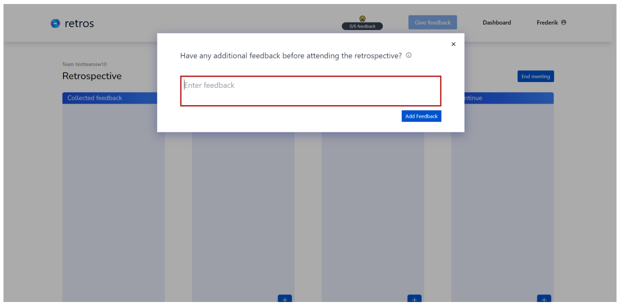
The initial idea for *creating friction*, was to use a color indicator around the feedback input text box, which should indicate to the user to input more characters. After reflecting on this implementation, it was rejected in favor of an implementation that focuses on feedback *quality* through semantic analysis of the input. Using a Python Natural Language Processing (NLP) library, *NLTK* [10], the platform automatically tokenizes and stems the user input, and by enabling multithreading it is capable of responding within milliseconds. This changes the color indicator such that it triggers based on a simplified semantic value instead of the number of characters. The color indicator will span across 4 different colors, transitioning from red to orange, orange to yellow, and finally yellow to green. The transition values, i.e. the semantic values causing color transitions, are predefined values which are discussed in Section 4. This nudge is enabled to increase the quality of developer feedback through the platform. The implementation of this nudge in our platform is illustrated in Figure 1.



We should discuss how we estimate tasks.

Figure 1. Implementation of the creating friction nudge.

The second confronting nudge, *throttling mindless activities*, is limited in that the platform is expected to rarely be used, i.e. when hosting SRs or voluntarily providing feedback. The nudge should not be overly eager, e.g. by prompting the users whenever they visit the platform. This nudge should be implemented to prompt users to submit feedback when joining the virtual SR unless they have submitted enough feedback beforehand. The prompt should ask the user if they want to submit any feedback before joining the meeting, whilst still giving them the option to not submit more feedback. This nudge is enabled to increase the quantity of developer feedback through the platform, however, the *creating friction* nudge is also enabled in this prompt. The implementation of this nudge in our platform is illustrated in Figure 2.



Have any additional feedback before attending the retrospective?

Enter feedback

Figure 2. Implementation of the throttling mindless activities nudge.

Building on top of the second confronting nudge, we plan to implement a reinforcing nudge of *instigating empathy*, which displays a smiley to the user with a happiness level reflected in the amount of feedback submitted by the user. The displayed smiley will change in a predefined order where the transition from one smiley to another depends on a predefined threshold value, i.e. amount of feedback submitted by the user. The predefined threshold value is discussed further in Section 4 along with what smileys were used. To exemplify this nudge, a user might be shown a sad smiley when no feedback is submitted, and after prompting their first feedback the smiley will change to a happier one. The smileys should be placed on the navigation bar of the platform, to always be visible to the user, and its purpose should be explained on hover. The implementation of the instigating empathy nudge can be seen in Figure 3. Finally, we implement the *raising the visibility of user actions* nudge as a visible counter for the user throughout the platform, which should be placed around the smiley from the *instigating empathy* nudge. This nudge should make their actions more noticeable, i.e. whenever they provide feedback this counter increases along with the "happiness" of the smiley. Both this and the instigating empathy nudge are enabled to increase the quantity of developer feedback through the platform.

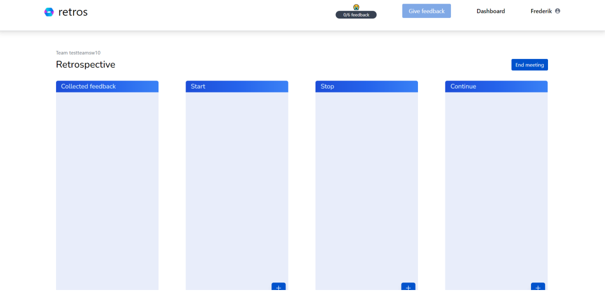


Figure 3. Implementation of the instigating empathy nudge.

Platform specifications

The nudging features are designed to be enabled by flipping a feature flag. The underlying platform is built to support developer teams conducting SRs and is based on the requirement specification from the prespecialization research. The platform supports the features illustrated in the flow diagram in Figure 4.

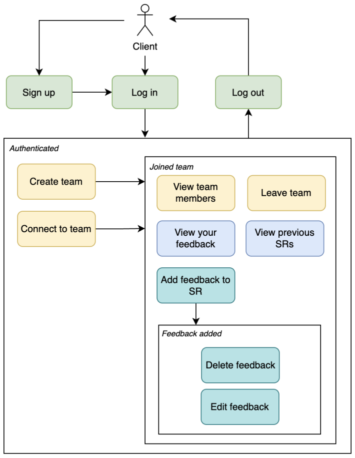


Figure 4. Overview of the platform features and workflow.

The client-side web application is built using NodeJS and React, and the server-side REST APIs are built using a microservice architecture with Java- and Python applications, deployed as containerized applications on Google Kubernetes Engine, and PostgreSQL relational databases as the underlying data storage.

4 Studying the Effect of Nudging in SRs

The grounds for this research is to understand whether nudging can be used to increase the quantity and quality of developer feedback in SRs. With a mixed-method approach, we collected quantitative data by implementing the *retros* platform in 2 developer teams while running an autonomous job that calculated the semantic values of all developer feedback added to the platform, followed by qualitative data collection through evaluation interviews with the participants.

Through a 2-condition within-subjects design we exposed the developer teams to either no nudging or nudging. We aim to define whether or not exposing the participants to nudging increases the semantic values of their feedback and the quantity of submitted feedback, as well as understand their experience of nudging through evaluation interviews.

Participants

The study is conducted with 2 developer teams of students studying Software Engineering at Aalborg University, which were contacted through study communication channels at Aalborg University. Both student groups are currently doing their Bachelor's degrees and in total count 8 participants (6 male and 2 female) between 20-23 years of age. All participants were both included in the 8-week quantitative data collection and the following qualitative data collection.

Prerequisites

Before initiating the study, we conducted pre-study preparatory meetings with all participating developer teams. During these meetings, we introduced the platform and its features, and created their accounts to reduce any friction during the study, and reduce the learning effect between the phases of the quantitative data collection.

Quantitative Data Collection

Previous research by Renaud and Zimmermann [12] shows that a hybrid approach of combining multiple nudges delivers significantly positive results compared to implementations with single nudges. Based on their findings, we design our study as a within-subjects design with 2 phases, where participants first will be exposed to the platform without nudging and then with nudging. The student groups are asked to use the *retros* platform to facilitate their SRs and incorporate it into their workflow. During the SRs we extract a semantic value from all developer feedback, i.e. a numerical value, with a label $nudge = true$ or $nudge = false$, to distinguish between feedback provided during phase 1 and phase 2. This data will be the quantitative data of the study. After closing phase 2, we conduct follow-up interviews with all participants individually to investigate the impact of the platform with and without nudges and collect qualitative data.

Phase 1: Nudging disabled

In this phase, nudging is disabled for all developer teams using the platform. The participants can facilitate SRs and provide feedback continuously throughout their Sprints, without any nudges from the platform. The duration of this phase was 4 weeks, i.e. all developer teams were in phase 1 for 2 Sprints (Sprint duration of both teams is 2 weeks), meaning we collected data for 2 SRs in this phase.

Phase 2: Nudging enabled

In this phase, we turn on the nudging features by enabling a feature flag. The platform includes the same set of features but is tweaked to introduce the 4 nudges; creating friction, instigating empathy, throttling mindless activities, and raising the visibility of user actions. The duration of this phase was again 4 weeks, providing data from 2 SRs.

In Section 3 we introduced the nudging features, without specifying threshold values for, e.g., the color indicator transitions of the creating friction nudge. We base these threshold values on data we collected from phase 1 of the quantitative data collection. In phase 1, 75% of data points (submitted feedback) had a semantic value of 9 or lower, 50% had a value of 6 or lower, and 25% had a value of 5 or lower. The creating friction nudge aims to increase this value, but the color indicator should transition relatively quick to not cause unnecessary user frustration, e.g. if the transition values are set to unfeasible values. Based on the percentiles from phase 1, the color transitions values were set as illustrated below in Figure 5.

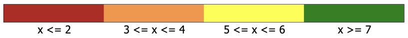


Figure 5. Transition values for the color indicator, where x is the semantic value.

The instigating empathy nudge aims to nudge users to submit more feedback. In phase 1, only 62,5% of users submitted on average 2 or more amounts of feedback during a SR, and only 25% submitted 6 or more. Based on this, we enforce a threshold value of 6 for the nudge, and the cutover from sad to happy should be 5 amounts of feedback. Again, we do not want to use overestimated threshold values and risk frustrating the user. The top threshold values should be feasible to reach for the users. The threshold values for the instigating empathy nudge are illustrated below in Figure 6.

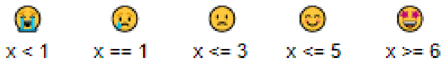


Figure 6. Smileys used for the instigating empathy nudge, where x is amount of submitted feedback.

The throttling mindless activities nudge also aims to increase the quantity of developer feedback submitted to the platform. Based on the threshold value for the instigating empathy nudge, this nudge should trigger until the user has submitted enough feedback to reach the threshold value of 6. This means that users will face a pop-up whenever they join a SR without having provided an amount of feedback equal to or greater than 6.

Qualitative Data Collection

The purpose of the qualitative data collection is to enrich our dataset with contextual insights to contribute to the quantitative data. If both datasets unveil the same findings, a mixed-method approach can reinforce the validity of the results.

We conducted individual semi-structured interviews with all participants to investigate their perception of whether nudging improves their feedback to SRs, and if they see themselves using a platform utilizing nudging in the longer term. We recorded the interviews for later analysis. Semi-structured interviews were favored, as we are determined on what we want to investigate. On the other hand, we want the participants to be able to speak freely on any topic, i.e. maybe they believe other nudges would be better, or that a different implementation of an existing nudge could be beneficial. The interview blueprint is listed in Appendix A.

We hosted the individual interviews after the quantitative data collection finished, meaning the users had 8+ weeks of experience using the *retros* platform. The interview participants included all 8 participants from the quantitative data collection, and each interview had a duration of 5-10 minutes.

5 Results

In this section, we present the results from both the qualitative- and quantitative data collection. The qualitative results are based on semi-structured interviews with the participants, and the quantitative results are based on data collected during a +8-week experiment using the developed *retros* platform.

Results from Quantitative Data Collection

From the platform, we analyzed 2 kinds of data based on the developer feedback submitted to the platform; semantic value pr. feedback and amount of feedback submitted pr. user pr. SR. The semantic value ties to the *quality* of developer feedback, and the amount of feedback submitted ties to the *quantity* of developer feedback. We divide the quantitative results into 2 phases, a phase 1, where nudging features were deactivated, and a phase 2, where all the nudging features previously presented were activated.

Phase 1: Nudging disabled

In phase 1, we collected 65 data points, i.e. the developer teams submitted feedback 65 times. The results of the semantic value pr. feedback are illustrated below in Table 2 along with the amount of submitted feedback pr. user pr. SR. The semantic value is calculated as the number of different tokens in the submitted feedback, where stop words such as "the", "a", "is", etc., are excluded. E.g. the sentence "Discuss how to estimate tasks." provides a semantic value of 3, whereas the sentence "I think we can improve how we are estimating tasks. We should discuss how to do this moving forward to find a common and reliable way of doing so." provides a semantic value of 12. In phase 1, we saw feedback scoring a low semantic value such as "*One-week sprints*" and "*Code reviews*". We also saw feedback scoring a high semantic value such as "*Trello works better than I remembered, but the physical Scrum board works better IMO, because it is always present, unlike Trello*".

Percentile	Semantic Value	1. Retro	2. Retro
10%	3.0	1.70	0.00
25%	5.0	2.75	0.00
50%	6.0	5.00	3.50
75%	9.0	6.00	5.75
90%	11.0	6.30	8.00

Table 2. Percentiles of semantic value scores pr. submitted feedback and percentiles of amount of submitted feedback pr. user. pr SR. for phase 1.

The minimum semantic value for submitted feedback was 1, and the maximum value was 17. The results show that 75% of the submitted feedback had a semantic value of 5 or higher, 50% of the feedback had a value of 6 or higher, and 25% had a value of 9 or higher. Only 10% of the feedback submitted during phase 1 had a semantic value of 11 or higher.

The lowest amount of feedback submitted by a user was 0 feedback, and the highest was 8, across 2 SRs. Comparing the first SR to the second, the lower percentiles saw a decrease in the number of submitted feedback, whereas the highest percentile saw an increase to 8 times submitted feedback. In the second SR, 25% of users did not submit any feedback.

Phase 2: Nudging enabled

In phase 2 we collected 90 data points. The results of semantic value pr. feedback and amount of feedback submitted pr. developer pr. SR are listed below in Table 3.

Percentile	Semantic Value	1. Retro	2. Retro
10%	3.0	1.8	0.0
25%	6.0	4.0	0.0
50%	8.0	5.0	3.0
75%	10.0	6.0	7.0
90%	14.1	8.2	12.6

Table 3. Percentiles of semantic value scores pr. submitted feedback and percentiles of amount of submitted feedback pr. user. pr SR. for phase 2.

The box plot illustrated in Figure 7 compares the amount of submitted feedback pr. SR pr. user of phase 1 and 2 in the study. The plus (+) sign indicates the mean values, and the bold line indicates the median values. The light blue boxes represent phase 1, and the light green boxes represent phase 2.

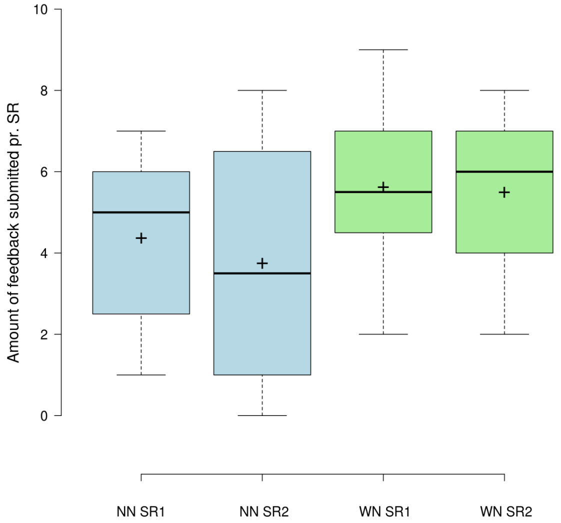


Figure 7. Comparison between the amount of submitted feedback pr. user across phase 1 and phase 2. *NN* refers to "no nudging", and *WN* refers to "with nudging".

Figure 7 illustrates that the lower quartile (Q1) has increased after nudging was introduced, along with the mean and median amount of feedback submitted, indicating that users increased the quantity of feedback submitted. We conducted a paired samples t-test to compare the amount of feedback submitted with and without nudging enabled. There was no significant difference in the data without nudging ($M = 8.1$ and $SD = 4.9$) and the data with nudging ($M = 11.1$ and $SD = 2.4$) as the t-test unveiled a p-value $p = 0.09$ (see Appendix C). This indicates that there is no difference in the amount of feedback submitted with or without nudging. However, comparing the total amount of feedback submitted between phase 1 ph_1 and phase 2 ph_2, we saw a 37% increase in data, as $ph_1 = 65$ and $ph_2 = 89$.

The box plot illustrated in Figure 8 compares the semantic value of the feedback submitted in phase 1 and 2 in the study. Figure 8 also illustrates an increase in the lower quartile, mean, median, and top quartile, indicating that users increased the semantic value of the feedback they submitted. Again, we conducted a paired samples t-test to compare the semantic value scores with and without nudging enabled. There was no significant difference in the data without nudging ($M = 7.2$ and $SD = 3.8$) and the data with nudging enabled ($M = 10.1$ and $SD = 3.3$) as the t-test unveiled a p-value $p = 0.06$ (see Appendix C). This again indicates that there is no difference in the semantic value scores with or without nudging. We still saw an increase of 40% comparing the semantic value scores before and after nudging was enabled.

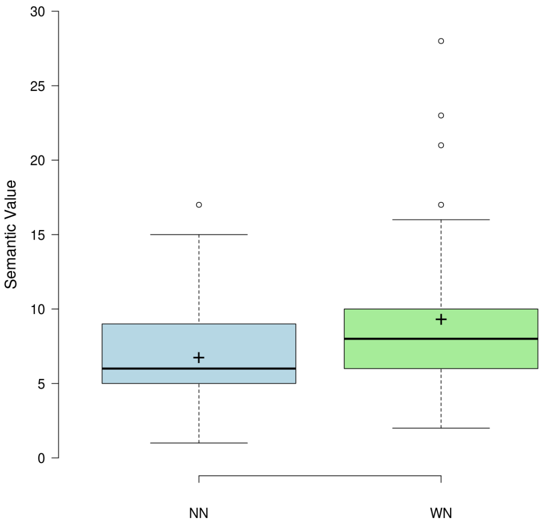


Figure 8. Comparison between the semantic value of submitted feedback across phase 1 and phase 2. *NN* refers to "no nudging", and *WN* refers to "with nudging".

Results from Qualitative Data Collection

We conducted semi-structured interviews with all participants after they closed their 2. SR with nudging enabled. In this section, we present the findings from these interviews, where P1-P4 denote participants from group 1, and P5-P8 denote participants from group 2.

Feedback Quantity Nudges

All participants described that the instigating empathy- and raising the visibility of user actions nudges made them strive towards submitting more feedback during their SRs. The participants described that they experienced a natural "gamification" during feedback collection, as they competed in achieving the happiest possible smiley, to which they used words such as "fun", "competition", and "maximum" to describe their interactions with the instigating empathy- and raising the visibility of user actions nudges. P1 described it as "*we had a good competition about getting the funniest smiley*". P8 mentioned that "*it became sort of a game. The counter gave me a bit more motivation, as I was always aware of how much feedback I had submitted*", however, to reach the maximum amount of feedback suggested by the nudge, P8 also expressed "*if I did not reach 6 feedback, I tried really hard to think about what other feedback I could write. Sometimes I succeeded coming up with new feedback and other times I just wrote something funny to reach the cap of 6 submitted* *feedback*". P8 mentioned that "*In our second SR when nudging was activated, I was not as motivated to reach the max count since I already knew which smileys would appear. However, it was nice to use the counter to track the amount of feedback I had submitted*", revealing a possible tendency of a drop-off in the efficiency from the nudge after the initial experience. P3 did not understand why the threshold value of the raising the visibility of user actions nudge was set 6, as they expressed "*I did not understand why the cap was set to 6. But when I reached the 6th feedback I thought it was fine and stopped.*". Almost all participants, except P5, mentioned that they never used the throttling mindless activities nudge. Participants P1-P4 simply closed it, whereas P6-P8 did not notice the nudge.

Feedback Quality Nudges

A few participants from group 1 expressed frustration with the creating friction nudge when trying to make the color indicator turn green. However, they then felt "*kinda over*" the nudge, and ignored the color of the input box. P1 expressed that "*Despite not actively trying to make the input box turn green, I still reflected more on the feedback I submitted*". P4 expressed that they also approached it this way, by first inputting their feedback and then reflecting on how it made them feel, before adding this to their feedback. P3 did not change behavior due to the creating friction nudge, but described that "*I noticed a difference in the feedback submitted by other team members*". P4 described that they perceived a general change in the feedback, where they usually would have to vocally explain their individual feedback to the rest of the group, whereas after the nudge was introduced, the feedback needed little or no further explanation and they could immediately discuss it. P2 also expressed that the submitted feedback was more detailed and needed less explanation during their SRs. Participants from group 2 did not experience any frustration with the creating friction nudge, and P5 described that "*I think it was easy to make the indicator green, and it was nice to get the recognition that my feedback was descriptive enough*". Like P3, P7 did not alter his behavior from the nudge, as they did not feel it was helpful nor made them reflect on their submitted feedback.

General Platform Perception & Usage

P1 and P4 described that their group did not use the feature to submit feedback before starting SR, which was intended to minimize friction for the users. Instead, they started a SR meeting and put feedback into the intended column (i.e. either "start", "stop", or "continue"). P1 stated that "*We moved away from submitting feedback before SRs, as it became a bit too general. Instead, we started a meeting and continuously joined it to submit feedback directly into the intended column. This also allowed us to see what feedback others had submitted*". Essentially, group 1 used the platform in an unforeseen manner, which added more friction than intended. P5 used the feature to add feedback during the Sprint and stated "*It was awesome being able to add feedback during the Sprint. This way I did not need to remember my feedback 2 weeks ahead. I used this feature a lot*".

Despite this, participants P1-P4 all enjoyed using the platform and would continue to do so, each with different reasons why. P1 mentioned that the nudges were "fun", and that the platform added structure to their SRs. P2 and P4 agreed with this, and P4 mentioned it was an improvement to previously used tools. P3 liked the structure added by the platform, but mentioned that "*Maybe I just don't understand the nudges*". However, P3 agreed with P1, P2, and P4, and would continue to use the platform. Participants from group 2, P5-P8, would also continue to use the platform, and P6 mentioned that "*If the platform continued to be available, we would definitely use it*". P6 described that "*We used the left-most column a lot, as we could iterate through each item of feedback one at a time, discussing which column it belonged to*", which was also the intended use of the platform. In extension to this, P6 also mentions that it would be nice to be able to add items to the left-most column, i.e. the "Collected Feedback", during the SRs, and not only before a SR is started.

6 Discussion

Through a mixed-method approach, we collected quantitative- and qualitative data throughout the study. Further investigating the results of the quantitative data from phase 1 unveiled a baseline of the quantity and quality of feedback submitted by the users. This data was used to manipulate the underlying threshold values of the different nudges but also serves as a comparison set with the quantitative data collected in phase 2 of the study. Previous work by Wang et al. [14] has unveiled influence of privacy nudges, where the authors also utilized a mixed-method approach combining the quantitative- and qualitative results to contextually describe the user behavior observed in the quantitative data.

Nudging Towards More Feedback

To nudge towards more feedback, we implemented the instigating empathy-, throttling mindless activities-, and raising the visibility of user actions nudge. The results from using these nudges will be discussed as well as the users' behavior interacting with the three nudges. Table 4 in Appendix B lists the amount of submitted feedback pr. user pr. SR.

We found that the three nudges increased the amount of feedback submitted when nudging was activated, compared to when nudging was disabled. The quantitative results are illustrated in Figure 7. The figure illustrates that the amount of submitted feedback has increased and is less diverse in phase 2 compared to phase 1. However, after discovering all the different smileys from the instigating empathy nudge, they lost interest in the nudge, but the raising the visibility of user actions nudge continued to remind them of how much feedback they had submitted. Perhaps if the combination of the shown illustrations of the instigating empathy nudge differed for each SR, e.g. a withered flower transforming into a lively flower or a caterpillar growing into a butterfly, it could sustain the users' interest in the nudge.

Even though the majority of the participants felt that the raising the visibility of user actions nudge motivated them to submit the amount of feedback it suggested, it did produce different behavior. For them to reach the threshold in the first SR, with nudging enabled, the participants sometimes added redundant or duplicate feedback, or when reaching the threshold they automatically stopped submitting additional feedback. This is not necessarily the desired behavior. The nudge should not make the user feel as if they must submit 6 items of feedback, or make them automatically stop submitting feedback when reaching the threshold. However, in the second SR with nudging enabled, they did not intend to reach the goal stated by the nudge anymore but merely used the nudge as a reminder of the amount of feedback they had submitted. This outcome is more desirable, which could be a motivation to try a different implementation of the raising the visibility of user actions nudge, where no limit is displayed to the user, only a counter.

As the nudges were only active for 2 SRs over 4 weeks, it can be difficult to conclude the users' behavior and satisfaction with the nudges. The time frame of this study was short, and it cannot be concluded with certainty that the participants would sustain their current behavior. Lee et al. [7] conducted a long-term study over 8 months on users' experience with a user interface that supports behavior change. From their study, they saw a shift in user satisfaction. In the first month, the users' satisfaction was high, before it dropped drastically in the 2nd and 3rd months. In the following months, it continuously rose slightly. Conducting our experiment over a longer period might yield different user experiences. This can also be said about the quantitative data illustrated in Figure 7. The data displays an increase in feedback during the phase with nudging activated. The amount of submitted feedback is more evenly distributed across users, and the minimum amount of submitted feedback has increased. The participants felt motivated and experienced a natural "gamification" during their SRs. However, if the users' experience with the nudges would change over time, other results would possibly emerge from the participant interviews. Notably, a paired samples t-test revealed no significant difference between the amount of feedback submitted with or without nudging enabled. We did not find these results unexpected, as our sample size $n = 8$ for the t-test was low. Perhaps a better suggestion of the effect introduced by the nudging feature is the 37% increase of submitted feedback from phase 1 to phase 2, and the difference in standard deviation (from $SD = 4.9$ to $SD = 2.4$) and mean value (from $M = 8.1$ to $M = 11.1$).

Nudging Towards Better Feedback

The creating friction nudge was implemented to encourage the participants to reflect on the feedback they submit and achieve higher quality feedback. The results obtained from the nudge and the experiences of the participants will be discussed. Table 5 in Appendix B lists the average semantic value of feedback submitted by each user.

This nudge produced mixed experiences among the participants, however, the majority of the participants expressed that while the nudge was activated, they either reflected more on their feedback or they noticed an improvement in the quality of the feedback from their peers. Compared to the three quantitative nudges, the participants' behavior was different when interacting with the creating friction nudge. In general, the participants did not feel that the green color indicator was the goal, but still reflected on the feedback they submitted. This shows that even though the nudge did not produce a sense of "gamification", the nudge still made the participants reflect more on their feedback.

The results collected from the nudge, illustrated in Figure 8, show an increase in the semantic value of the submitted feedback. Higher semantic value does not necessarily guarantee an increased quality of the submitted feedback. The underlying NLP engine does not evaluate the semantic value based on the detail of the feedback, but rather intelligently counts the number of words submitted, i.e. it does not distinguish between words such as "Monday" and "overwhelmed". However, the majority of the participants felt that the nudge made them reflect more when submitting feedback and that they spent less time discussing the meaning of the submitted feedback. This could indicate that the nudge had a positive effect on the quality of the submitted feedback.

Agapie et al. [1] conducted a study using a similar halo color indicator nudge. The goal of their study was to test whether users would type longer queries using the halo nudge. The halo nudge would transition from red to blue as their queries became longer. They ran two experiments, with 61 participants in the first, and 84 participants in the second. First, they tested whether the halo nudge had an effect. In the second experiment, they tested different conditions of the halo nudge, e.g. static colors versus transitioning colors. They found the halo effective in nudging the participants towards longer queries. Our results from a paired samples t-test revealed no significant difference between the semantic value of submitted feedback with or without nudging, which was expected due to the low sample size $n = 8$. Due to the similarity of the study conducted by Agapie et al. [1], it could indicate that the creating friction nudge had a real effect, and running a study with the appropriate dimensions, i.e. a longer study with more participants, could yield significant results. Also, another measurement of the success is the overall increase of semantic value scores between phase 1 and phase 2, where the score rose by 40%, and the difference in mean values (from $M = 7.2$ to $M = 10.1$).

Limitations

This study contained limitations that may affect the validity of the paper. The study was conducted as part of a master thesis project, and therefore its duration was limited to a few months. This, combined with a mixed-method approach requiring quantitative data, required the study to be initiated as early as possible to collect as much data as possible, which led to the study having only 2 student groups as participants and a sample size of 8 participants. Ideally, we would have wanted the participants to be experienced software developers from the industry, as this may have provided deeper insights into the practical applications of nudging. Additionally, had we conducted interviews continuously throughout the study, we may have gotten a better understanding of the users' interaction with the platform. However, this may have been too cumbersome given the time constraints. We would also have wanted to pilot test the nudges before initiating the study, which was not feasible due to the time constraints. This meant that the design of some nudges did confuse the participants, which could have affected the users' behavior towards the nudges. Finally, the nudges produced no data regarding their usage, which further led to a lack of knowledge regarding the users' behavior towards the nudges and therefore may have influenced the findings, i.e. we could have more confidence in regards to which nudges were most effective.

Future Work

Based on the results and limitations of the study, we encourage other researchers to extend this exploration of using nudges in the work environment. Our findings display a clear opportunity of using nudging in the work environment to better extrapolate team member data from SRs. Our study has limitations, as mentioned previously, which cloud the validity of our results. In the future, more research in this field could contribute further to the development of nudges within SRs.

Study Design

We applied a mixed-method study design, collecting both quantitative- and qualitative data for approx. 2 months. As previously mentioned, other researchers [8] found that the user satisfaction with nudging features peaks after 1 month, and then heavily declines after 2 months. Ideally, our study should be extended to expose the participants to nudging features for more than 1 month, to also capture the possible drop-off in satisfaction.

This study investigates the implementation of confronting-, reinforcing-, and social influencing nudges. Caraban et al. [3] also present *facilitating*-, *deceiving*-, and *fear* nudges, and previous research by Renaud and Zimmerman [12] has revealed that fear nudges improved password complexity of users, where they explicitly presented a possible loss to the users during the interaction. Any extension of this study should test out multiple and different combinations of nudges through different measurements.

Platform Design

During our participant interviews, we discovered further improvements to the platform in general, which are not necessarily related to the nudges. In general, the participants expressed that the platform would benefit from features such as

- *Timer.* A timer during SRs to be able to time cap the meeting.
- *Action Items.* An extra column, or page, or something else entirely, to support the creation and follow-up on action items derived from the SRs.
- *"Collected Feedback"-column update.* Allow users to input feedback into this column. It should be considered to not allow users to put feedback directly into the start, stop, or continue columns, as this enforces a conversation about the feedback and in which column it belongs. However, this might add unnecessary friction for some users.

It could be interesting to include a measurement that explains the effect of each implemented nudge, i.e. if one nudge is 90% more effective than others, it might not be relevant to include the other nudges.

7 Conclusion

This paper was motivated by the lack of software developer engagement in SRs, prior work by Thaler and Sunstein [13] introducing nudging in HCI research as a way of altering people's choice architecture, and research by Caraban et al. [3] presenting different types of nudges mapped into the HCI design space. The study described in this paper investigates the effect of a set of nudges on the quality and quantity of developer feedback submitted in SRs. The results indicated that the introduction of nudging in SRs caused greater participant reflection, resulting in a positive effect on both the quality and quantity of the submitted feedback.

Acknowledgments

We thank the student groups who participated in the study. A special thank you to our supervisor Niels Van Berkel for supervising this master thesis project, and the continuous dedication and guidance.

References

[1] Elena Agapie, Gene Golovchinsky, and Pernilla Qvarfordt. 2013. Leading People to Longer Queries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 3019–3022. https://doi.org/10.1145/2470654.2481418

[2] Nicolai Bjerring, Frederik Jacobsen, and Christopher Soerensen. 2021. Augmenting the Facilitation of Sprint Retrospectives. *9th Semester Software, Aalborg University* (December 2021), 1–70.

[3] Ana Caraban, Evangelos Karapanos, Daniel Gonçalves, and Pedro Campos. 2019. 23 Ways to Nudge: A Review of Technology-Mediated Nudging in Human-Computer Interaction *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300733

[4] digital.ai. 2021. *15th State of Agile Report*. digital.ai. https://digital.ai/resource-center/analyst-reports/state-of-agile-report

[5] Brendan Fisher. 2008. Richard H. Thaler and Cass R. Sunstein: Nudge: Improving Decisions About Health, Wealth, and Happiness: Yale University Press, New Haven. In *Yale University Press*, Alastair Walker, Rory V. O'Connor, and Richard Messnarz (Eds.). Springer International Publishing, Cham, 532–545.

[6] BJ Fogg. 2009. A Behavior Model for Persuasive Design. In *Proceedings of the 4th International Conference on Persuasive Technology (Persuasive '09)*. Association for Computing Machinery, New York, NY, USA, Article 40, 7 pages. https://doi.org/10.1145/1541948.1541999

[7] Sang-Su Lee, Youn-Kyung Lim, and Kun-Pyo Lee. 2011. A long-term study of user experience towards interaction designs that support behavior change. In *CHI '11 Extended Abstracts on human factors in computing systems (CHI EA '11)*. ACM, 2065–2070.

[8] Sang-Su Lee, Youn-kyung Lim, and Kun-pyo Lee. 2011. A Long-Term Study of User Experience towards Interaction Designs That Support Behavior Change. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 2065–2070. https://doi.org/10.1145/1979742.1979909

[9] Christoph Matthies, Franziska Dobrigkeit, and Alexander Ernst. 2019. Counteracting Agile Retrospective Problems with Retrospective Activities. In *Systems, Software and Services Process Improvement*, Alastair Walker, Rory V. O'Connor, and Richard Messnarz (Eds.). Springer International Publishing, Cham, 532–545.

[10] NLTK Project. 2021. *Natural Language Toolkit*. digital.ai. https://www.nltk.org/

[11] Adam Przybylek and Dagmara Kotecka. 2017. Making agile retrospectives more awesome. *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems* 11 (2017), 1211–1216. https://doi.org/10.15439/2017F423

[12] Karen Renaud and Verena Zimmermann. 2019. Nudging folks towards stronger password choices: providing certainty is the key. *Behavioural Public Policy* 3, 2 (2019), 228–258. https://doi.org/10.1017/bpp.2018.3

[13] Richard H. Thaler and Cass R. Sunstein. 2008. *Nudge: Improving Decisions About Health, Wealth, and Happiness* (1st ed.). Yale University Press New Haven & London, USA.

[14] Yang Wang, Pedro Leon, Alessandro Acquisti, Lorrie Cranor, Alain Forget, and Norman Sadeh. 2014. A field trial of privacy nudges for Facebook. *Conference on Human Factors in Computing Systems - Proceedings* (04 2014). https://doi.org/10.1145/2556288.2557413

A Experiment Participant Interviews

In this appendix we list the questions and introduction to the semi-structured interviews held with participants of the experiment.

We introduced the interview as a semi-structured interview, and told participants that they should simply think out loud when asked a question, and share how they felt about the platform both with- and without nudging. The interview questions are focused on mainly on how the participants perceived their own contributions, i.e. did they actually feel that they gave more consideration on their feedback. We want to clarify if they noticed the nudges, and if they ever felt "pushed" to give another contribution or better describe their experiences during SRs. The questions for the interview are listed below.

- Did the platform influence your SRs?
  - If yes, how and why do you think it did?
  - If no, why do you think it did not?
- Did the smiley, counter, and/or popup features influence how much feedback you submitted in SRs?
  - If yes, how and why do you think it did?
  - If no, why do you think it did not?
- Did the color indicator feature influence the detail of your submitted feedback?
  - If yes, how and why do you think it did?
  - If no, why do you think it did not?
- What are your thoughts on continuing to use the platform for your SRs?

B Data Collection

In this appendix we list the collected data used for plotting. All data collected was based on a user submitting feedback. From this data, i.e. a user submitted feedback, we received the following entities:

- **id**: auto-generated id to identify different feedback entities.
- **data**: the actual text the user submitted.
- **category**: the column feedback is placed in, i.e. start, stop, continue, or none.
- **retrospective id**: the id of the retrospective the feedback belongs to.
- **nudge**: if true, this feedback was collected when nudging was enabled.
- **created by**: the id of the user who submitted the feedback.

Using these entities we could process how much feedback each user submitted in each SR. In the representation illustrated below in Table 4, the user identifiers are replaced with integers between 1-8.

	No nudging		**With nudging**	
User	**SR1**	**SR2**	**SR3**	**SR4**
1	6	2	6	7
2	6	8	8	7
3	6	3	4	5
4	7	8	6	5
5	3	4	5	3
6	4	5	5	8
7	2	0	2	7
8	1	0	9	2

Table 4. Amount of submitted feedback pr. user pr. SR.

Using the *data* entity, an automatic job calculated the **semantic value** entity of each submitted feedback. Using a Python script, all submitted feedback was mapped to the user it was given by and split into before and after nudging was enabled. The average semantic value of each list is illustrated below in Table 5.

User	**No nudging**	**With nudging**
1	4.917	8.722
2	7.714	8.444
3	6.0	6.111
4	5.333	6.652
5	1.0	12.8
6	11.0	13.625
7	13.5	15.0
8	8.0	9.231

Table 5. Average semantic value from submitted feedback pr. user with and without nudging enabled.

C Data Processing

In this appendix we list the calculations for the paired samples t-tests.

To calculate the p-value using a paired samples t-test we firstly calculate the sample standard deviation of the differences s_{diff}, which we will need to calculate the estimated standard error of the mean s_x.

$$s_{diff} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x^*)^2}{n-1}}$$

, where x^* is the mean of the sample and n is the sample size.

$$s_x = \frac{s_{diff}}{\sqrt{n}}$$

Finally, we can calculate the statistic t-value t using the sample mean of the differences x_{diff}, i.e. the mean of all the differences from set A and B, and the estimated standard error of the mean s_x:

$$t = \frac{x_{diff} - 0}{s_x}$$

From the t-value t, we use a p-value calculator to find the p-value p using $n - 1$ degrees of freedom, where $n = 8$, a significance level of 0.05, and a two-tailed configuration.

Amount of feedback pr. user

Using the data from Table 4, we can calculate whether there is a significant difference between the platform with and without nudging. Using the formulas above, we get

$$s_{diff} = 4.4078$$

and

$$s_x = \frac{4.4078}{\sqrt{8}} = 1.55839263506$$

and

$$t = \frac{3-0}{1.55839263506} = 1.92506043247 = 1.925$$

Using $t = 1.925$ we get a p-value of $p = 0.0956$, i.e. the results are not significantly different.

Semantic value of feedback pr. user

Using the data from Table 5, we can calculate whether there is a significant difference between the platform with and without nudging. Using the formulas above, we get

$$s_{diff} = 3.777$$

and

$$s_x = \frac{3.777}{\sqrt{8}} = 1.33537115627$$

and

$$t = \frac{3-0}{1.33537115627} = 2.24656642156 = 2.25$$

Using $t = 2.25$ we get a p-value of $p = 0.0.059$, i.e. the results are not significantly different.