BACHELOR THESIS

Sending signals in open source: Evidence from a natural experiment

submitted by

LUKAS MOLDON

The present work was submitted to the

Chair of Computational Social Sciences and Humanities

within the

Faculty of Mathematics, Computer Science and Natural Sciences

at RWTH Aachen University

March 17, 2020

Advisor:

Dr. Johannes Wachs

First Supervisor

Prof. Dr. Markus Strohmaier

Second Supervisor

Prof. Dr. Jan Borchers

Abstract

A significant part of the new economy uses open source software, which is freely provided by OSS developers on various platforms. GitHub is one of the most important platforms for open source software development and sends important signals of developer activity via profile pages. In this thesis we analyze behavioral effects of a profile design change on GitHub in 2016, removing counters of a user's ongoing and all-time longest unbroken streak of daily contributions. We show that these "streak" counters were important for a part of GitHub's community and that removing the feature changed the behavior of developers in different ways. Besides a general decreased interest in streaking after the change, we see that women were less affected by the feature than men, suggesting that they are less susceptible to gamification. We find other social impacts, including a decrease in weekend work and network effects through the site's social network. We conclude with a discussion of the benefits and risks of the streak feature and draw lessons for the application of gamification in online platforms.

The corresponding code for all computations in this thesis is available under:

`https://github.com/lukasmoldon/GHStreaksThesis`

Contents

1

Introduction

1.1 Motivation

Open source software developers play an important role in our economy and society since a significant part of the new economy is built on the contributions they freely share [21, 52]. A survey by the Linux Foundation from 2018 found that 72% of all surveyed companies use open source software for non-commercial or internal reasons and 55% utilize OSS for commercial products [20]. Because of this reliance it is important to better understand why developers contribute to open source and what incentives they face. One important and understudied aspect of the open source software ecosystem is how platform design itself influences behavior. In this thesis we explore how changing the ability of developers to signal information about their contributions can influence or even steer their behavior. We exploit a design change on GitHub, the leading online platform for collaborative open source software development, made in 2016 to measure effects of gamification elements. This unannounced change allows us to test hypotheses about the motivations of developers using observational data.

1.2 About GitHub and streaks

As a leading platform for open source software development with more than 32 million registered users[1] and a global Alexa rank of 75[2], GitHub is of major importance to the whole IT sector. Originally founded as *"Logical Awesome LLC"* in 2008, GitHub is now a subsidiary of Microsoft, which acquired the platform for $ 7.5 billion in 2018.[3] GitHub provides a platform for using the Git distributed version control system. Developers use GitHub to collaboratively write software and to follow the behavior of other projects and developers. 2.1 million organisations and busi-

[1] GitHub user search: `https://GitHub.com/search?q=type:user&type=Users` (September 23, 2019)
[2] Alexa rank: `https://www.alexa.com/siteinfo/GitHub.com` (September 23, 2019)
[3] Microsoft has acquired GitHub for $7.5B in stock: `https://techcrunch.com/2018/06/04/microsoft-has-acquired-GitHub-for-7-5b-in-microsoft-stock/` (September 23, 2019)

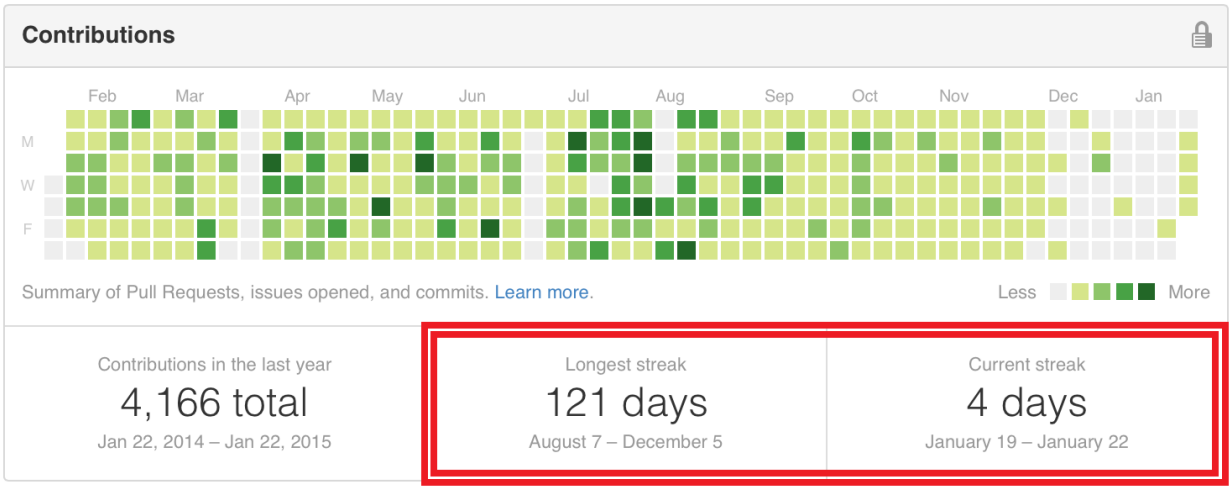

Figure 1.1: *Example of streak counters taken from a user profile prior to May 19th, 2016 (sourced from* `https://zachholman.com/posts/streaks/`*). The design change removed the two highlighted counters, signaling the lengths of a user's longest and current streaks.*

nesses use GitHub, including the likes of Google, Facebook, NASA, PayPal and IBM.[4] GitHub is also an important hub for the open source software community [15]. A survey by JetBrains from 2019 found that 73% of all surveyed developers (55% of them contributing to open source) regularly use GitHub as a Version Control Service [29].

Today, the platform hosts more than 125 million repositories with 1.3 billion commits. Every registered user has his/her own public profile, including the possibility to provide personal information like the full name, a short biography, the user's location, a company he/she is working for, or a personal website. Besides the idea of building a large online community, the profile is an important resource for job search. There is even an "Available for hire" checkbox in the profile settings for users and a standalone platform called "GitHub Jobs" for companies, which search for employees with specific skills. In general GitHub profiles are important representations of the identities of software developers, with consequences for their careers [11, 49]. In this sense, participation on GitHub has significant impact on labor market outcomes. As of 2019, GitHub is positioning itself as an important economic actor in the open source community through "GitHub Sponsors" - a mechanism for grateful users to tip their favorite open source software developers [40].

GitHub user profiles also contain important signals of developer activity. The contribution graph is a calendar of the past 365 days reporting the user's daily activity, see Figure 1.1. Previously several statistics about the user's contribution patterns were reported below the calendar including

[4]GitHub front page (not signed in): `https://GitHub.com/` (September 23, 2019)

how many days in a row the user had made a contribution and his or her all-time longest streak. These features were removed without advanced notice on May 19, 2016, while the contribution calendar is still present today. This change made some users angry, suggesting that these statistics were important to some users.[5]

From a research perspective, this unannounced design change presents an opportunity to measure the effects of the counters on user behavior. We consider the removal as a quasi-experiment, and compare user behavior right before and after the change, avoiding many of the barriers to causal interpretation that observational studies of online platforms usually suffer from [34]. In the following we define research questions and hypotheses on signaling and developer activity that we plan to test using this event. We review the relevant literature and summarize previous related findings. We then describe our data source and filtering. Afterwards, we present our findings: Comparing behavior right before and after the design change, we observe statistically significant changes in several kinds of activities. We conclude with a summary of our findings and their limitations, potential areas for future work, and implications for the design of collaborative platforms and the open source community.

1.3 Research questions

The angry reaction of some users to the removal of streak counters suggests that they were important to at least a part of GitHub's community. We first want to examine if there was in fact a statistically significant change in streaking, which we define as a long uninterrupted streak of daily activity, right before and after removing the streak feature from GitHub:

RQ 1: Did the design change affect user behavior (especially streaking) significantly?
RQ 2: If there is a change in user behavior (RQ 1), how and why has it changed?

There are many different potential reasons for streaking. Streaks of significant length could be used as signals of personal achievement and dedication on the user's public profile. A user with strong streak statistics may have an advantage in the labor market, as streaks represent high activity, which is associated with experience/knowledge [44]. This leads us to our first hypothesis, that without having the ability to signal streaks, users generally lost the incentive to engage in streaking and stopped maintaining their personal streaks after the change (H1). We would expect to see that ongoing longest streaks were abandoned after the design change and that new long streaks became more rare (H2). We compare the daily share of users having a (significant) streak over time, zooming in on the time around the design change and employing statistical methods to test H1. For H2 we analyze the change in survival rates of long streaks across the change.

[5] See GitHub issues `https://GitHub.com/isaacs/GitHub/issues/627` and `https://GitHub.com/dear-GitHub/dear-GitHub/issues/163` (September 23, 2019) Backup (screenshots) available at: `https://GitHub.com/lukasmoldon/GHStreaksThesis/tree/master/saves`

In contrast to the ongoing streak counter, the maximum streak counter represents a qualitatively different kind of achievement. The removal of this second feature could have resulted a decreased interest in beating previous personal bests (H3), as users lost the ability to see and signal current personal record streaks on their profile. Thus, we filter our database for maximum streaks and analyze the share of users beating their past records and examine the maximum achieved streak in a user's lifetime.

Recognizing that GitHub users are a heterogeneous population, we are also interested in identifying which subpopulations were most significantly effected by the design change. In other words: We would like to identify groups of users who put significant effort into streaking and signaling.

RQ 3: Are different groups of users more impacted by the change, and if so, what are their defining characteristics?

Past research suggests that men are more likely to respond to gamification elements on online platforms [18, 35]. We expect men to have longer streaks on average compared to women before the change and hence being effected more significantly by the change (H4). Besides the separation in gender, some developers may be contributing at work - typically Monday to Friday - and others in their free-time. At-work contributors could be influenced less by the design (H5), as developers working rather on nights and weekends may be more interested in signaling [38]. The user's origin could also influence this context and streaking in general, as different countries have different cultural attitudes towards long periods of work and signaling (H6). By repeating the same statistical analysis on data from users from these specific groups, we can observe differences in signaling behavior in different populations of the userbase.

Streaks may also have been used by users as an intrinsic motivational tool. For example a developer learning a new programming language may set the goal to work on a project in that language every day for 30 days. We hypothesize that the design change could have influenced this behavioral signpost (H7). We test this by observing users participating in goal based streaking communities and analysing achievers of long streaks of round lengths (i.e. 100 days).

RQ 4: Did the design change have further (non-streaking) behavioral effects on GitHub?

In order to keep long streaks "alive", users have to contribute at inconvenient times such as on the weekends, on holidays, or in the evenings. We would expect the overall amount of commits on weekends to fall (H8), which will be tested with a regression discontinuity design method on the share of weekend activity.

Finally we note that signalling requires other users to notice personal achievements. We thus hypothesize that the removal of signals changed behavior through GitHub's social network. Before the change, developers could compete with their connections and reflect on the contribution patterns of people in their network. Afterwards, observations of the streaks of neighboring users became more indirect. Specifically, we claim that the synchronization of streaking behavior fell across the design change (H9).

1.4 Related work

Both the streak counter for the current streak and the counter for the longest streak can be considered as badges [3]. Badges are tokens that introduce a *"gamification"* aspect, an umbrella term for the use of (video) game-design elements in non-gaming contexts like GitHub [13]. Research shows that this can affect user behavior in two different ways: First, it can lead to an increased participation on the platform. Second, it can change the user behavior on the site and could even be used by the platform owners for steering user behavior [3]. This will be relevant for discussing the question of why GitHub implemented the feature in the past and why they removed it in 2016.

Past work emphasizes the importance of enjoyment-based intrinsic motivation for OSS developers in particular [32], which could be encouraged by gamified elements. For example, repository badges on GitHub are optional gamified elements, which incentivize higher activity for best quality assurance and keeping projects up-to-date [48]. On Stack Overflow, another important platform for OSS developers, past research finds that badges have a strong steering effect, as user activity on the site increases significantly right before a badge is awarded [25]. Users are also interested in collecting points and tokens in casual settings, for example in the world of online chess [2]. This suggests that some users may use gamified elements to set goals and to stay on target.

Besides being strong incentives, badges can also function as a credentialing system, as they summarize the users' achievements, hinting at the users' overall experience and skills [3]. As mentioned in the introduction, GitHub operates the standalone platform "GitHub Jobs", which links the users' GitHub profile to the job application. This attributes a new meaning to badges on GitHub, as a potential employer reviews the user's linked profile and uses this information to decide who to hire [44]. Previous work on GitHub has also considered differences in behavior and success across subpopulations. For example, women are underrepresented and at disadvantage in teams [41], but their code acceptance rate is higher, conditioned on hiding their gender [47].

Previous work shows that the social network of OSS developers is important for their success and the resulting popularity of their collaborative projects [26, 50]. Thus, it is interesting to analyze a possible connection between the social follower network and the streaking behavior of all users on GitHub.

2

Data extraction and computation

The following chapter describes the process of extracting data and creating different databases for filtered groups of users. The corresponding code for all computations in this thesis (including data analysis) is available at: `https://GitHub.com/lukasmoldon/GHStreaksThesis` .

2.1 Data extraction from GHTorrent

Our primary data source is GHTorrent[1], a continuously updated database of information retrieved from the GitHub REST API [24]. For this research we are using the latest provided MySQL database from June 2019. The data set contains 32.5 million users, 125 million projects, 100 million opened issues and 1.368 billion unique commits. The next step was to filter this dataset to address our research questions and to facilitate complex computations on the whole data.

We only want to consider users active around the time of the change with a significant amount of contributions. From originally 32.5 million users, we discarded all users who did not have a commit in a non-forked repository (17.3 million remaining). We also removed users with more than 100 invalid commit timestamps to filter "cheating" users and bot accounts (see 3.3.3). Bots make up a significant number of contributions, so it is important to filter them out carefully [14]. Then, we focused on users with at least 100 commits, a "USR" type (private account with one single owner - no organisation), not being marked as fake by GHTorrent, and with available location coordinates, leaving 433,138 users. Figure 2.1 shows the number of created accounts over time of our observed user group, more than 85% of them joined GitHub before the design change. Coordinates are required for identifying the users timezone, as every timestamp gets transformed and saved in UTC-0 by GitHub, but streaks are evaluated by local time zones. For example, without knowing that a user lives in San Francisco, his/her commit at 8PM local time on a Monday would be falsely evaluated as a commit on a Tuesday (3AM in UTC-0). These coordinates are computed by GHTorrent, using the location text field on every user profile and the OpenStreetMap API.

[1] `https://ghtorrent.org/` (February 19, 2020)

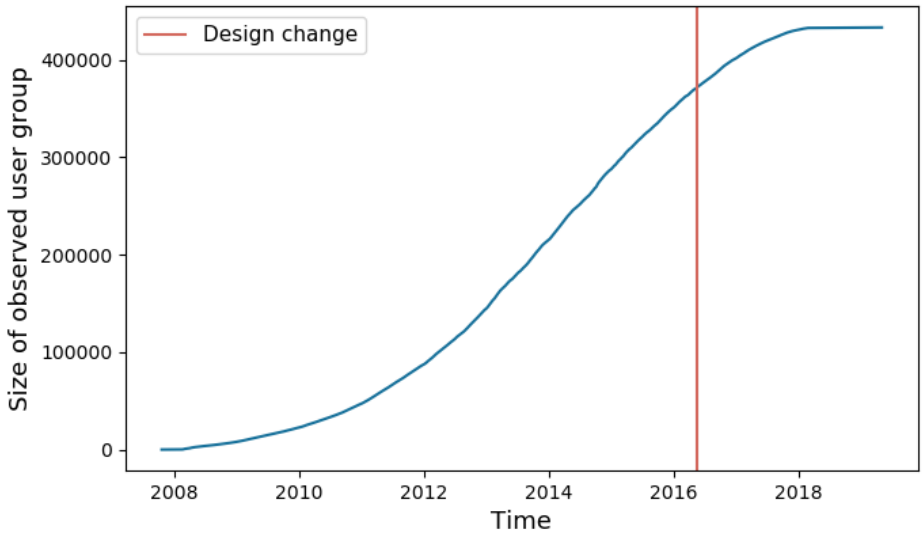

Figure 2.1: *Number of created accounts over time for the observed user group*

The next goal was to create a database of all streaks ever made by this set of users. The streak feature counted three types of contributions: commits, pull requests and issues can all count as a contribution under some requirements [2]. For instance, contributions only count if they are associated with a standalone (non-forked) project. For pull requests and issues, we had to check if they were made in a forked repository and filter them out. However, because 48 million projects represent a forked copy of a corresponding standalone project, commits are assigned to projects 6.252 billion times. Whenever a project gets forked, all commits of this origin standalone project get duplicated and assigned to the forked project copy, too. So here we had to discard commits to forks which were never merged to the original project. We created filtered databases for each contribution type for all observed users. As a result, we are analysing the data of 433,138 users with over 290 million valid contributions (including 12.8 million issues and pull requests). In the last step, all contributions were sorted by time and user. We computed the resulting data set of streaks (start, end) assigned to the corresponding user ID.

2.2 Data verification

In order to ensure the correctness of our created databases and the consistency of the GHTorrent data with the original GitHub data, we generated 10,000 random queries from each computed database for the GitHub API. We translated the internal user IDs to the pair of the login name and the account creation date, since users can change their names on GitHub over time and could take the past username of some other user. Over 99% of our queries matched with the API response,

[2] `https://help.GitHub.com/en/articles/why-are-my-contributions-not-showing-up-on-my-profile` (September 28, 2019)

except small errors in timestamps. The remaining failed queries are a result of untraceable users and actions, most likely reflecting accounts or actions deleted after June 2019.

We also tested the opposite direction to ensure that no original data is missing in our databases. Again we created the same amount of queries randomly from the GitHub API (only for our observed users) and tried to find the response in our database. Many of the queries are not part of our database, as we filtered out all contributions in forked repositories and only have data until June 2019. Regardless of filtered activities, further 1.5% of all queries failed. We assume two main reasons for this inconsistency: First, contributions in private (hidden) projects, which became public afterwards, are available today but are not part of our database. However, the streak feature only counted contributions in public projects, so it is even important to filter out past private activities. Second, inconsistencies such as incorrect entries and minor holes in data collection are unavoidable. We assume that this error is distributed uniformly over time and is equal before and after the design change.

2.3 Gender detection algorithm

In our research questions we hypothesized that women are less influenced by gamification elements like the streak feature. To test this idea we infer the gender of each user in our observed set. The GHTorrent database does not contain any information about the user's gender, as there is no option to display gender on GitHub profiles. However, users can state their full name on their GitHub profile, which is not part of GHTorrent's data. We use this information to perform dictionary-based geolocated gender detection. These dictionaries record the relative frequencies of given-names by gender at birth, recorded by national or regional public administrations. Location can have a significant impact on gender inference, for example the first name "Andrea" is associated with men in Italy, but most commonly used by women in other countries like Germany [51]. We note that we are making significant simplifying assumptions about gender: that it is a static and binary phenomenon and that classification errors from name-based gender inference are balanced.

In the first step, the gender detector crawls the GitHub API to obtain the full name of each observed user. For about 8% of the users we found no full name, further 21% stated a name that could not be associated with a name in the dictionary. Second, the script requests the user's home country name, using the given coordinates and the OpenStreetMap API [3]. Following previous work [51, 35] to infer gender from usernames, we use the python package Gender-Guesser[4] to infer the user's gender. Gender-Guesser uses a dictionary [5], where frequencies of first names and

[3] `https://wiki.openstreetmap.org/wiki/API_v0.6` (February 21, 2020)

[4] `https://GitHub.com/lead-ratings/gender-guesser` (February 19, 2020)

[5] `https://raw.GitHubusercontent.com/lead-ratings/gender-guesser/master/gender_guesser/data/nam_dict.txt` (February 19, 2020)

gender are saved for different regions. These regions do not always represent a single country and do not cover the full world. We wrote a translator which matches an OpenStreetMap country result with the corresponding region (or returns no match). With the translated region and the first name as input, Gender-Guesser classifies the gender with different categories of certainty. For further analysis we only keep results with the highest certainty (male/female). Thus, 53% of the users (230k) are inferred as men or women. The share of women is 7.3% (17k), in line with similar estimates of the presence of women in the OSS community and GitHub specifically [51, 12, 43]. We must note a further limitation: as research indicates existing biases against identifiable women on OSS platforms [8, 47] and a tendency of women to provide less likely job relevant information on online user profiles [1], women may be more likely to hide their gender in their profiles.

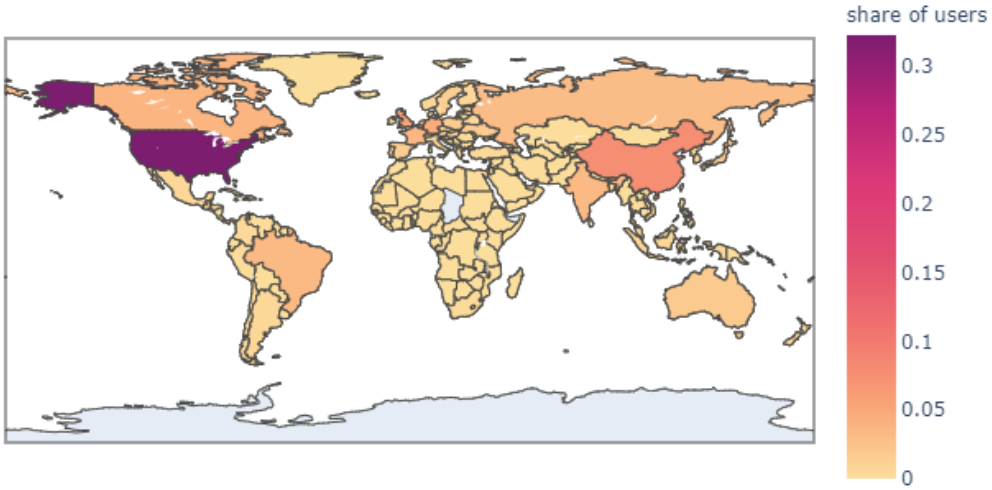

Figure 2.2: *Country distribution of observed users, emphasizing strong influence by developers from the USA.*

Gender is not the only imbalanced characteristic in our sample of users. Figure 2.2 visualizes the country distribution of all 430k observed users, using the computed country names from the gender data. The map emphasizes strong overrepresentation of US American developers in the data with an overall share of 32%, followed by China (8%), UK (5.4%), Germany (4.7%) and Canada (3.8%). Moreover, nearly 65% of all users are from Western Europe or North America, in contrast to around 1% from Africa. The overrepresentation of Western countries is also a common and well studied phenomenon in OSS [16, 22]. There is also a significant relationship between a user's geographic location and the likelihood that his/her contributions are accepted to GitHub projects [42].

3

Data analysis and results

3.1 General development of streaking behavior

We now turn to our analysis of user behavior. We study different aspects of streaking behavior before and after the platform design change. First we introduce the chapter with general findings about the share of streaking users. Afterwards we consider the distributions of streaks starting on Mondays. We then we discuss results of a streak survival rate analysis and conclude with a different way of examining streaks, the streak density.

3.1.1 Share of streaking users

We first would like to test whether changes in behavior across the design change were indeed statistically significant (RQ1). In order to answer this question, we computed the share of all observed users having a streak with a minimum length of 20, 60 and 200 days for each day. For computing these values we count streaks for each group (20, 50, 200) from the day they passed the threshold t of a group, not from the day they were started. This highlights short temporary effects on streaking, like holidays or service outages. Moreover, for each day we divided by the size of the observed user group on $t-1$ days before (t is the threshold of each group). This respects the fact that a user can only have a streak of length t on a specific day if he joined more than $t-1$ days ago. The resulting plot (Figure 3.1) shows that the largest drop of streaking users within 3 years happened immediately after the 19th of March in 2016 and that the overall share of streaking users decreased afterwards, confirming H1. However, users having streaks longer than 200 days did not change their behavior directly after the change, which seems to contradict H2. This will be investigated in further analysis when observing lifetime streak records.

Other incidents are visible, which result in temporary drops. We can not explain all events, which happened within these 3 years, as there may be many factors that impact streaking. We marked Christmas and the US Independence Day (July 4), which shows that users tend to abandon streaks across major holidays. Drops at the Independence Day underscore the previously mentioned sig-

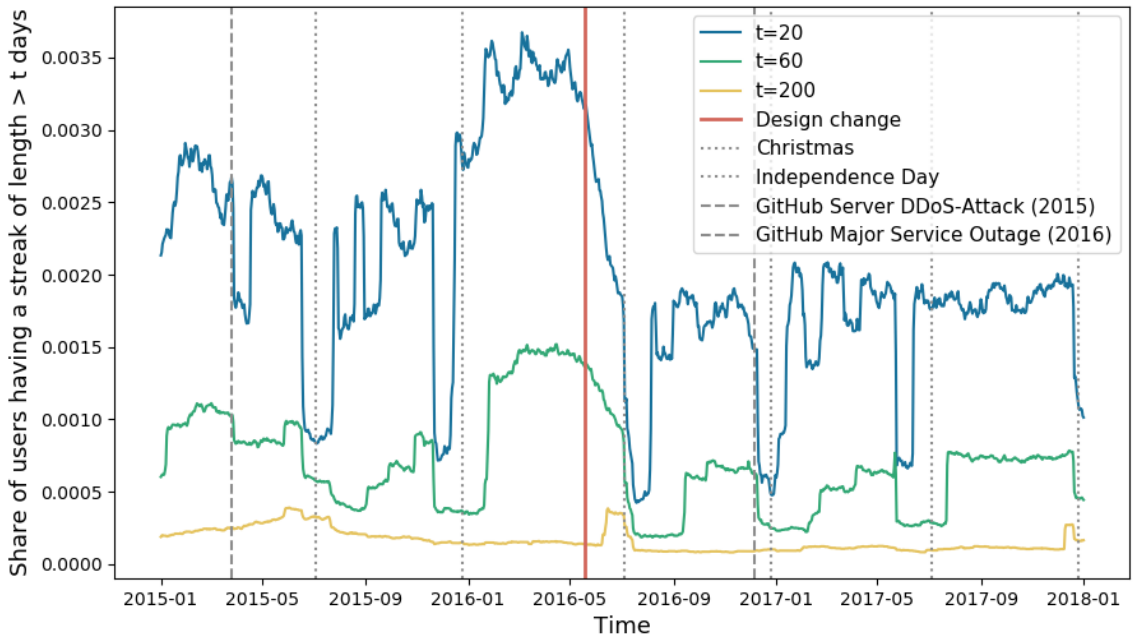

Figure 3.1: *Share of users having a streak of length > t days for $t \in \{20, 60, 200\}$: Largest drop happened right after removing streaks from GitHub (red line). Users tend to abandon their streaks during the holiday season (dotted lines) and streaks are affected by server outages (dashed lines).*

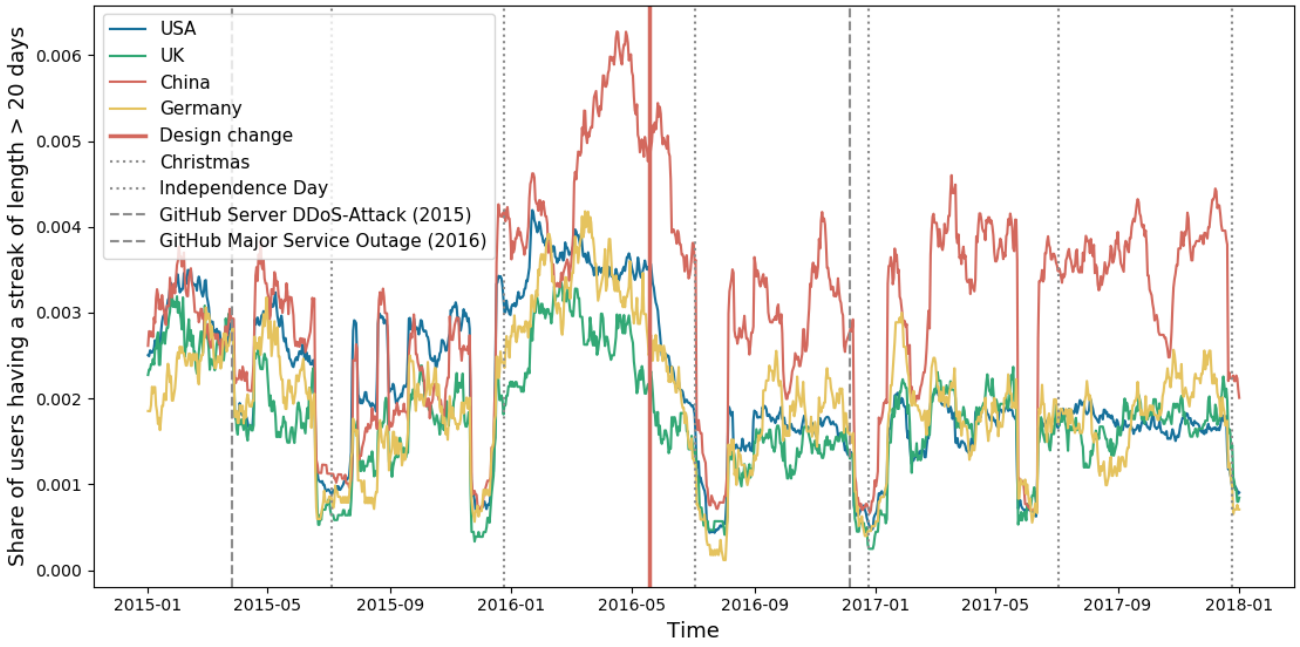

Figure 3.2: *Share of users from different countries having a streak of length > 20 days: While western countries are affected equally by the design change, users from China continue streaking afterwards on a similar level.*

nificant presence of US American users on GitHub. There are other types of incidents, which can effect streaking: In March 2015 GitHub suffered with a DDoS-Attack [1], which resulted in an drop of streaking users with length 20 and 60. A major service outage in 2016 resulted in a similar impact on streaking.

Figure 3.2 shows the same graph for $t = 20$ only, but for users from different countries. While western countries are affected equally by the design change, Chinese users continue streaking on a similar level (with one temporary drop across Independence day). One explanation could be that Chinese developers often have significantly more demanding working hours than their counterparts in the Western world [27], shedding light on H5 and H6. This interpretation is supported by recent protests against long working hours on GitHub by Chinese developers and their supporters [54, 33].

Mann-Whitney-U test for H1 To test the statistical significance of the change in streaking behavior across the design change, we computed the Mann-Whitney-U statistics for two samples of streaks. While sample x contains streaks starting in the three weeks before the design change, y contains those starting three weeks after the change. $|x| = 359,189$ and $|y| = 352,437$ implies $U_{max} = |x| \cdot |y| = 125,591,493,593$. The resulting statistics are $U_x = 63,092,489,716.5$ and $U_y = 63,499,003,876.5$, which means that the distribution of sample y has a lower mean compared to sample x, since the sum of all ranks is higher (and the first rank is for the highest streak length). The resulting p-value is $p = 0.0037$ indicates significance at $p < .01$, confirming a significant change.

3.1.2 Monday streaks

We now shift our focus to the analysis of streaks starting on different days of the week. It is interesting to observe only those streaks starting on Monday for several reasons. We expect users who use GitHub at work to start streaks on Mondays, relating to H5. Second, we can get a first impression of the survival rates of streaks over weekends (H8) and third, we can focus on the longest streaks for H2. In the following we analyze streaks starting on Mondays in the first ten weeks in 2016 (before) and 2017 (after the design change). Figure 3.3 shows the average distribution of streak lengths starting from the observed Mondays in 2016. It shows that the majority of the Monday streaks ends on the same day, so that 50% of the users, who made a contribution on these Mondays made none on the following day. For the rest of the week we can see a decreasing trend. However, we have a peak on Friday which is even higher than the values on Wednesday or Thursday. This means that the chance that a Monday streak ends on the next Friday is significantly higher than that it ends on the day before and is even higher compared to Wednesday. As suggested, it seems employees have a high impact on this statistic. Another reason could be the phenomenon of a decreased number of commits after Mondays, described by Gousios

[1] `https://www.GitHubstatus.com/history?page=19` (September 29, 2019)

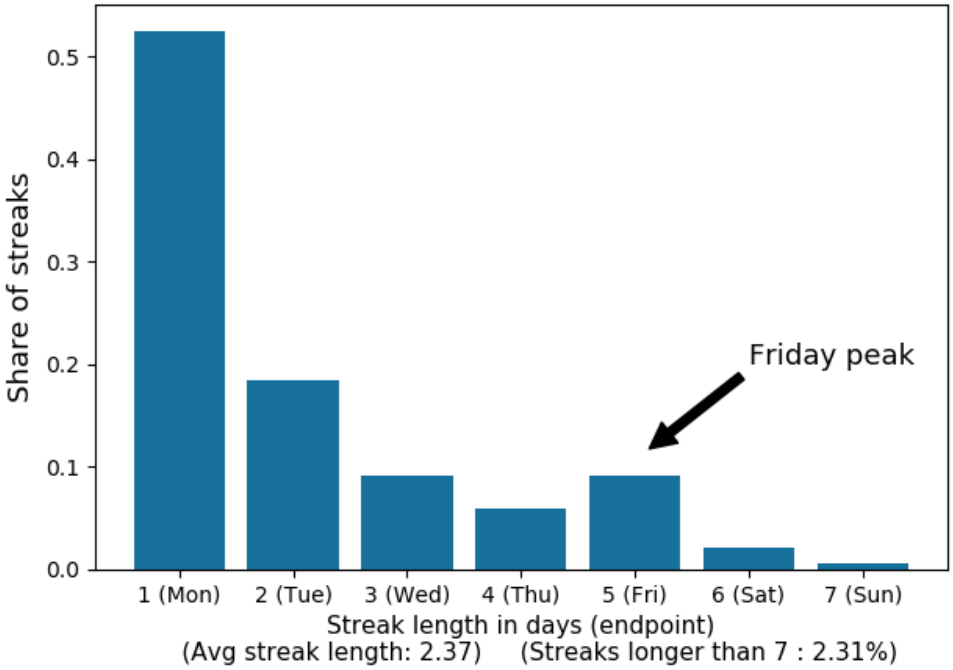

Figure 3.3: *Streak length distribution for streaks with length < 8 starting at the first 10 Mondays in 2016 (avg. values): More than 50% of the streaks end on the first day, a significant part of streaks ends on the following Friday mainly caused by employees.*

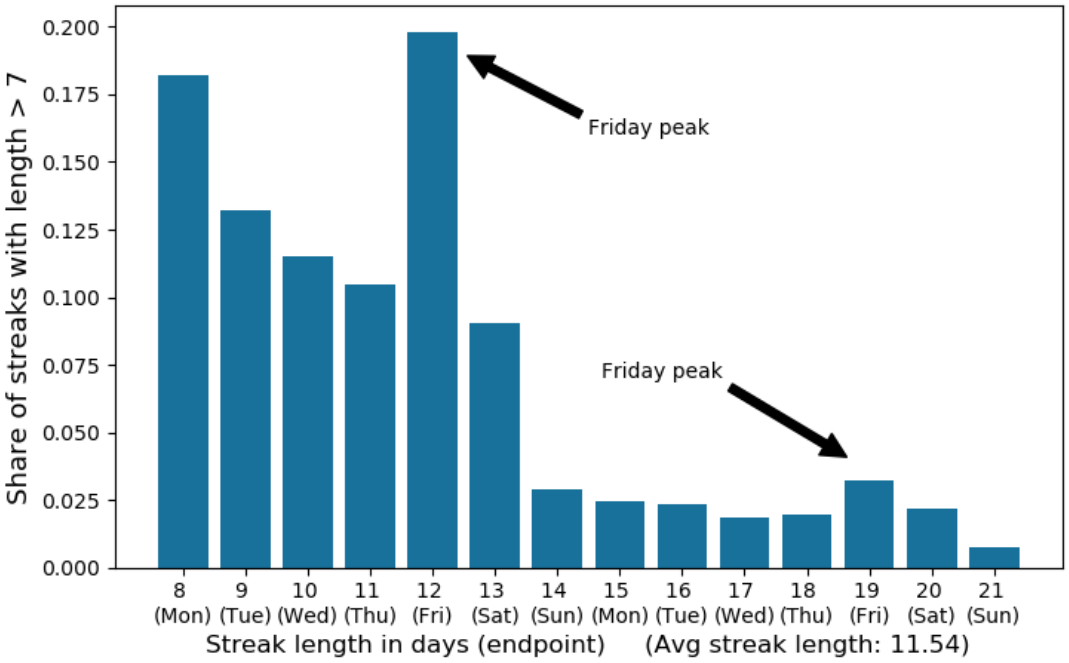

Figure 3.4: *Streak length distribution for streaks with length 8 - 21 starting at the first 10 Mondays in 2016 (avg. values): Significant peaks are at both Fridays, which again shows that employees have a high impact on these statistics.*

and Spinellis as “The Monday Effect” [23]. For streaks starting at the same Mondays but with a length over 7, we can observe a second peak at length 12 in 3.4, which represents the Friday the week after. There is even another Friday peak at length 19, which could be caused by employees working over one or two weekends and then taking a break or the general decision by users to prefer other (offline) activities on weekends.

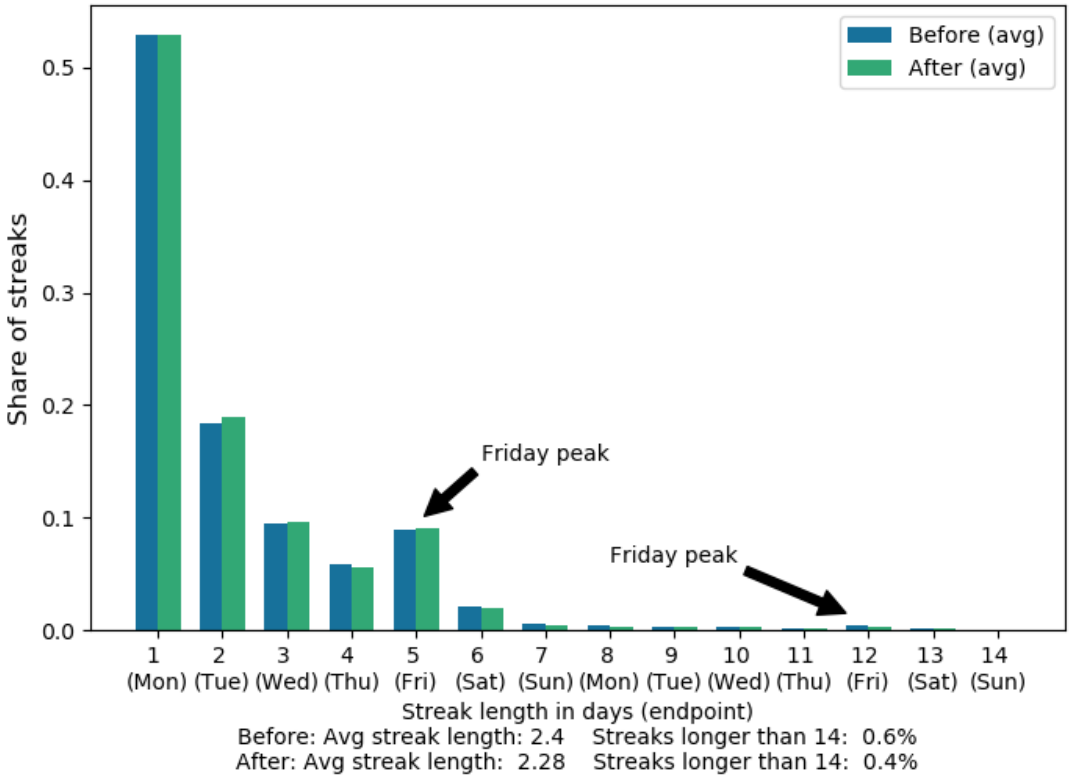

Figure 3.5: *Compared streak length distribution for streaks with length < 15 starting before (2016) and after (2017) the design change on Mondays in Jan. & Feb. (avg. values): Friday peaks remain after the change and the distributions are nearly identical for the first 2 weeks. But the share of longer streaks drops by a third to 0.4%, which is illustrated in Figure 3.6.*

Since we are focusing on the design change, the next step is a comparison between Mondays before and after GitHub removed streaks from profiles. Figure 3.5 compares the distribution of the observed Mondays in 2016 and 2017. Friday peaks remain after the design change and the distributions seem to be nearly equivalent, suggesting that employees are streaking naturally - even without having the ability to signal their streaks (H5). However, the share of streaks longer than 14 days, decreased from 0.6% to 0.4%. The difference increases when considering longer streaks. Figure 3.6 shows that the design change caused a decreased chance for having (Monday) streaks longer than 34 days. Nearly all streaks, which started on Mondays in 2017, ended after 34 days at the latest, while more than 8% of the Monday streaks being longer than 2 weeks in 2016, had a length over 54 days. Moreover, the chance of observing streaks longer than 100 days decreased significantly from 4.36% to under 2%, supporting H2.

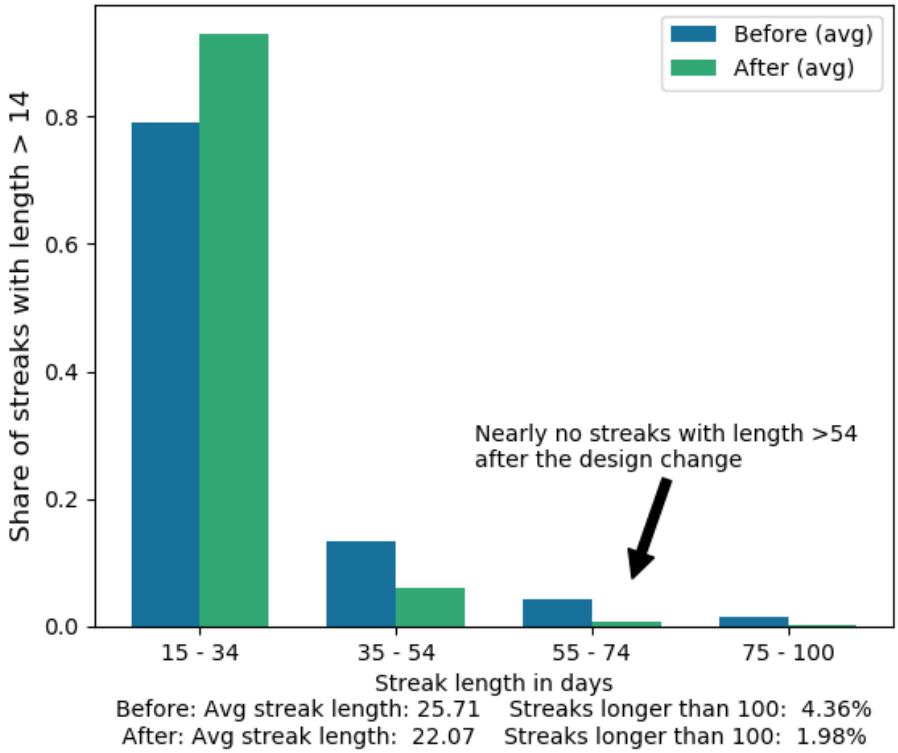

Figure 3.6: *Compared streak length distribution for streaks with length > 14 starting before (2016) and after (2017) the design change on Mondays in Jan. & Feb. (avg. values): The design change is followed by a lower chance of observing streaks longer than 34 and nearly no streaks with a length over 54.*

When analysing specific Monday streaks around the design change, we observe similar patterns: The probability that a 7 day streak survived to 14 days dropped significantly from 15% to 1 % after the change, as Table 3.1 shows. This is evidence for H2, as very few long streaks are started after the change (limitation on Mondays for now). Figure 3.5 is also evidence for H5, since there is no real change in the first 14 days starting from a Monday, where employees mainly start their unintentional streaks, which do not seem to be affected by the design change. We also repeated these calculations with streaks starting on other weekdays in both years and found a similar but decreased Friday peak effect (e.g. the probability that streaks starting on Tuesday end on Friday is nearly the same as for ending on Thursday).

Starting date	avg length	P(length >7)	P(length >14 | length >7)
2016/04/18	2.38	0.52%	15%
2016/04/25	2.29	0.40%	10%
2016/05/02	2.24	0.40%	11%
2016/05/09	2.36	0.43%	7%
2016/05/16	2.33	0.45%	9%
2016/05/23	2.30	0.39%	9%
2016/05/30	2.27	0.40%	6%
2016/06/06	2.27	0.31%	5%
2016/06/13	2.24	0.27%	1%
2016/06/20	2.28	0.35%	3%

Table 3.1: *Comparison of streaks, which started on a specific Monday (Date). Lower average streak lengths after the change represent reduced streaking. Decreased probability of streaks longer than 14 indicates a loss of interest in maintaining long streaks.*

3.1.3 Streak survival

Another way of measuring streaking behavior is to analyze survival rates of streaks, which represent the daily probability that a streak will be continued to the following day. We found no significant decrease in daily survival rates right after the design change and assume two reasons: First, the abandonment of long streaks happened naturally every day before and after the change, as streaks do not survive endlessly. We already saw in Figure 3.1 that the abandonment of streaks is spread over several weeks after the change, reactions on the GitHub Forum indicate that at least some users hoped for a comeback of the feature. This hinders the separation of users who changed their behavior and quit streaks because of the design change, and those who naturally ended their streaks. Second, users who are interested in signalling but were not streaking around the change are not part of this analysis, while users who constantly maintain streaks of lengths over 200 days (Figure 3.1) are omnipresent in the calculation. Similar to Anderson and Green [2], who analyzed the probability of quitting streaks of online chess games around an individual's previous high score, we additionally computed the probability of continuing a streak when approaching the length of the previous maximum streak. Once again we found no significant change in behavior. However, the chess game setup did not punish platform absence (only losing games), while on GitHub a single day offline resets the streak counter. Thus we infer that users try to keep their streaks alive as long as possible, even after beating previous bests.

To bypass the described interfering effects on survival rate calculations, we analyzed the survival of streaks starting in the same week over time. Figure 3.7 shows the average share of streaks surviving at least x days, while each line represents one of the first 10 weeks in 2016 and 2017 (before and after the change). The separation of the two colours highlights that streaks before the design change survived longer on average. Especially for the length of 20 to 40 days we observe a

gap between the lines, which illustrates a decreased probability of observing these streaks after the change by 0.5%. This fits to H1 and H2, as less streaks get maintained over longer time periods after the change. However, the figure also shows that long streaks a rare in both years, which will be further analyzed in 3.2.

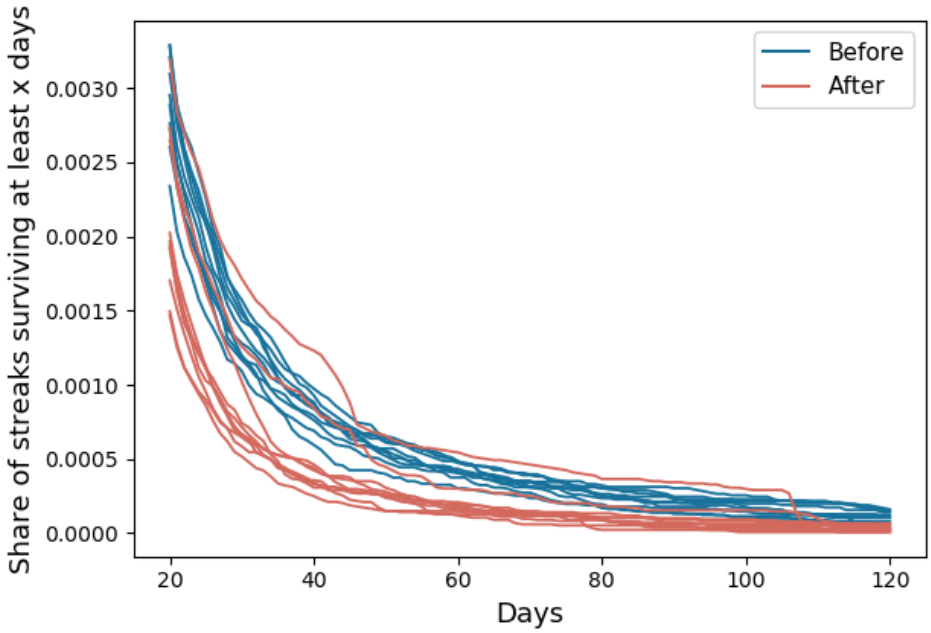

Figure 3.7: *Weekly share of streaks surviving at least x days (lines represent first 10 weeks in 2016 and 2017 as streak start): Streaks before the design change survive longer on average.*

3.1.4 Streak density

So far we analyzed streaks regarding their length or their survival over time, which only measures if there was at least one contribution on each day or not. However, each contribution means more effort for a user, while additional contributions on the same day are not rewarded by the streak feature. It is imaginable, that streaks contained less contributions at the end of a streak's lifetime with a present streak feature, as users try to keep their streaks alive with the minimum required effort. In the following we analyze how contributions and especially days with only one contribution are distributed over streaks. The null hypothesis is that contributions occur randomly in streaks and that they are uniformly distributed. We also check the change in distribution of one commit days across the design change.

Figure 3.8 shows the distribution of contributions over all streaks one year before/after the change with a minimum length of 30 days. To keep streaks of different lengths comparable, we split the streak's lifetime on a relative scale from 0% (starting day) to 100% (ending day) into ten deciles (bars). Thus, a bar represents different total amounts of days for different streaks, but always 10%

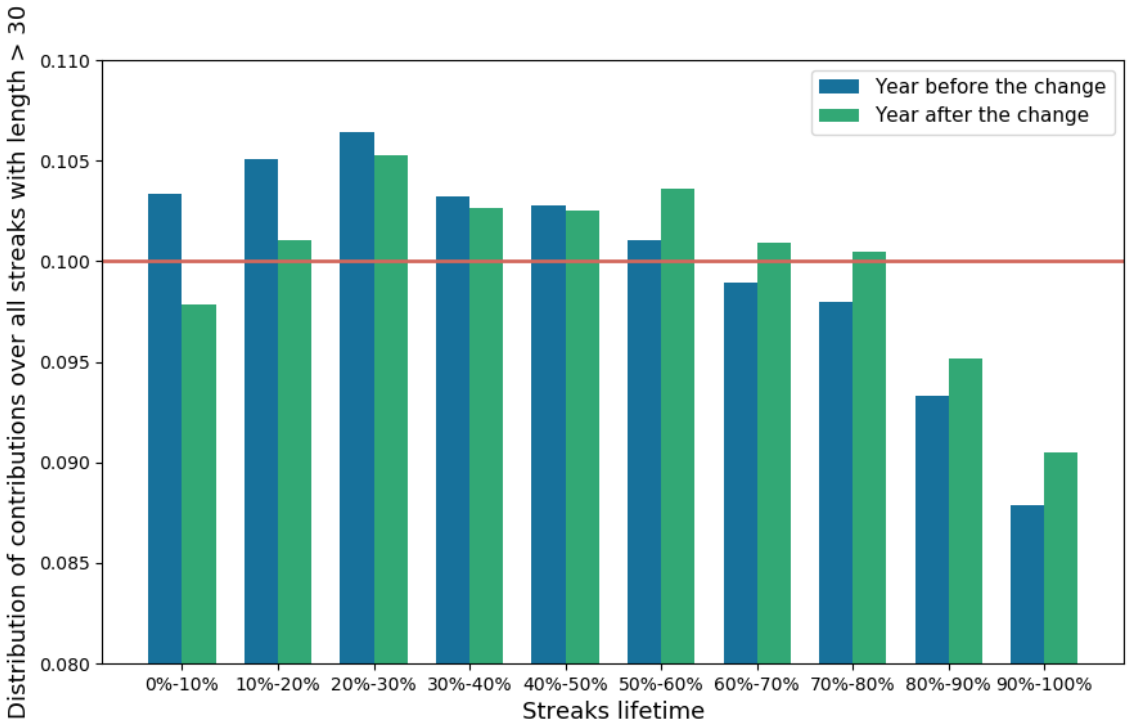

Figure 3.8: *Distribution of contributions over all streaks one year before/after the change with a minimum streak length of 30. Contributions are not uniformly distributed (red horizontal line) and decrease in the last quarter of a streak's lifetime.*

days of the total length of each streak. The red horizontal line represents the null hypothesis. Both distributions differ from this uniform distribution, as there are 1-2% less contributions in the last quarter of a streak. For each decile (except 50%-60%) we observe a distribution after the change, which is closer to the null hypothesis. To test for statistical significant difference from the null hypothesis, we calculated the Chi-squared test[2] of the distributions under the uniform distribution. The Chi-squared test yields $\mathcal{X}^2_{before} = 32.01$ with $p_{before} = 0.0002$ and $\mathcal{X}^2_{after} = 39.43$ with $p_{after} = 0.00001$ indicating significance at $p < .01$.

Figure 3.9 illustrates the share of days with only one contribution over the streak's lifetime, while focusing on streaks with a minimum length of 60. Note that (different to the previous plot) the share is calculated within each decile and does not represent a distribution with $\sum_x p(x) = 1$ over the whole streak. Besides a 2.6% higher overall share of one contribution days (OCDs) before the change, we once again observe a non-uniform distribution before the change and an increasing chance of observing an OCD with an increasing streak lifetime. In the last decile of a streak we observe the highest probability of observing OCDs, suggesting that users indeed tried to maintain streaks with a minimum effort at the end of a streaking period. The distribution after the change

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chisquare.html` (February 26, 2020)

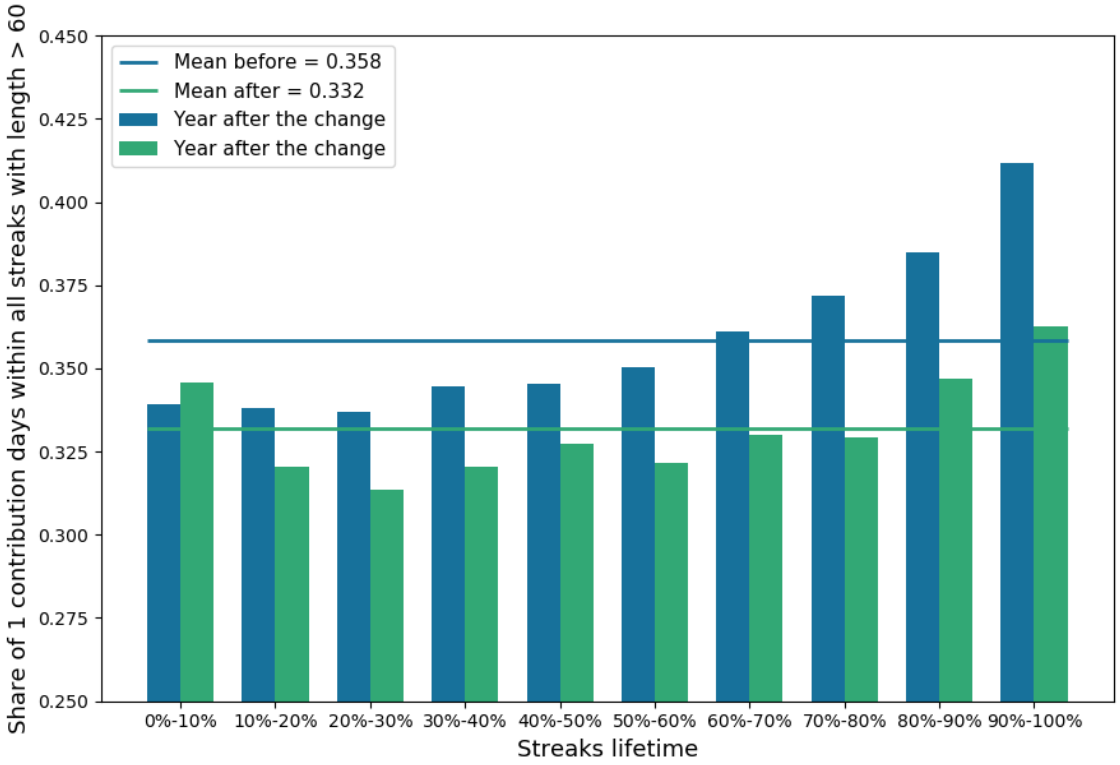

Figure 3.9: *Share of days with one contribution over all streaks one year before/after the change with a minimum streak length of 60. One-Contribution-Days are not uniformly distributed and have a higher share at the end of a streak. After the change this tendency weakens.*

is closer to its own mean (horizontal line) and more uniformly distributed. But how much single contributing can be expected at random, and how can we determine the statistical significance of these patterns?

To analyze how often OCDs occur naturally at random, we calculate a z-score (standard score) for each user for the year before and after the change as follows. Given active days D with at least one contribution and the total amount of contributions C of all days of a user, we test how often OCDs occur randomly: First, each active day receives exactly one contribution. Second, the remaining $|C| - |D|$ contributions are assigned randomly to all days. Third, we repeat the test 1,000 times. Now we can compare the average number of randomly occurred OCDs $avg(\#OCD_{random})$ with the observed value $\#OCD_{data}$. The z-score

$$Z = \frac{(\#OCD_{data} - avg(\#OCD_{random}))}{\sigma_{random}}$$

represents the user's deviation from the null hypothesis, that contributions occur randomly within a streak, which implies a random occurrence of OCDs (σ_{random} denotes the standard deviation of OCDs in the randomisations). While $Z = 0$ means an identical representation of OCDs in the real data of a specific user and the randomisations using the user's C and D, $Z > 1.96$ indicates an overrepresentation and $Z < -1.96$ an underrepresentation with p < 0.05. Figure 3.10 shows the distribution of z-scores for all users for the year before the change. The plot is cut at $x = 40$, but there exist observations up to $Z_{max} = 743$. While the red line denotes the threshold for users having a signficant overrepresentation of OCDs, the yellow line marks the mean z score for all users. Thus, we observe a significant overrepresentation, compared to the random model. Figure 3.11 represents the same distribution as cumulative distribution function, including the year after the change. While 19% before and 16.4% of all z-scores are above 10, 2.1% before and 1.6% of the observations after are higher than 40 and not part of the plot. Both distributions represent a significant overrepresentation of OCDs, with a mean z-score of 7.54 before and 6.82 after the change. Overrepresentation decreased after the change and a Kolmogorov-Smirnov test[3] on the cumulative distribution of z-scores confirms significant statistical difference between both distributions with $p < 0.01$. We also note that the significant over-representation of OCDs after the change suggests support for the hypothesis that activity in software development is bursty - meaning highly correlated in time [9].

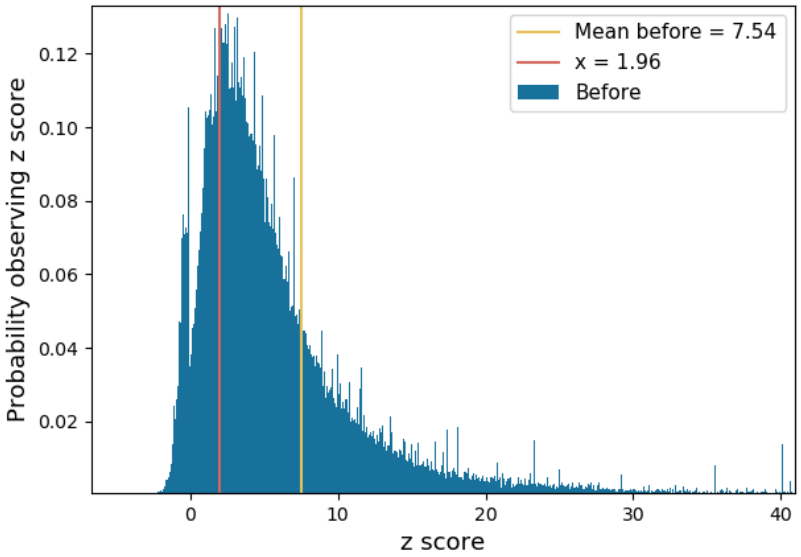

Figure 3.10: *z-score distribution (PDF) for the year before the design change: The majority of z-scores is higher than 1.96, which represents a statistical significant overrepresentation of One-Contribution-Days compared to the random model.*

[3] https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.ks_2samp.html (February 26, 2020)

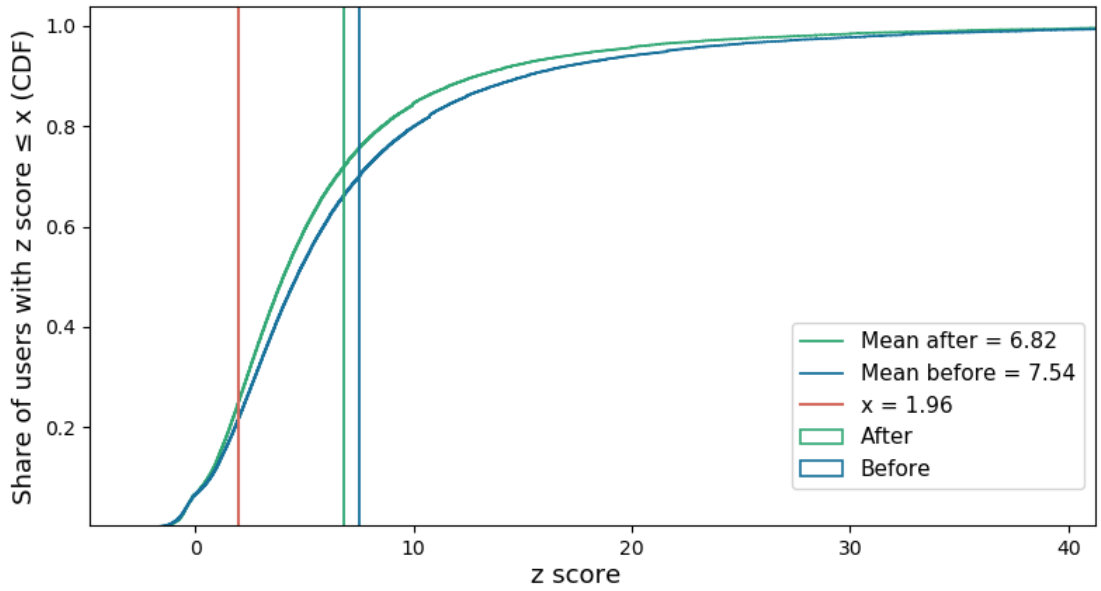

Figure 3.11: *Cumulative distribution function of z-scores before and after the design change: Both curves represent a statistical significant overrepresentation of One-Contribution-Days. A lower mean and a curve closer to p=1 for x < 40 indicates a decreased over-representation after the design change, the change is statistically significant with p<0.01 (KS-test).*

Overall before and after the change we observe behavior, which differs significantly from our null hypothesis. However, the distribution of contributions is closer to a uniform distribution after the change, days with only one contribution occur less often at the end of a streak and appear generally less deviant from the random model for all users without the presence of the streak feature. These findings support again H1 and H2 and suggest that streaking users focused on activities, which got rewarded by the gamified feature.

3.2 Influence of the maximum streak badge

In the following we want to analyze how the maximum streak badge in particular may have influenced user behavior (H3). Thus, we first study retrospectively the development of maximum streaks over time and then analyze the maximum streak record in each user's lifetime.

3.2.1 Beating personal records

As already mentioned, previous work emphasizes the importance of past bests as reference points, as users (for example in online chess games) increase their effort as they come close to beating their old personal records [2]. To carry out this analysis we created a new streak database. Starting with a first significant streak with a minimum length of 15 days (if existing) for each user and we only keep following streaks, which beat the latest maximum streak. Thus we recreated the

sequence of maximum streak lengths on each user's profile over time.

Figure 3.12 shows the daily share of users having a new ongoing maximum streak from 2015 until 2017. Similar to previous figures in 3.1, we observe the largest drop right after the design change and holidays/service outages seem to affect the values again. Moreover we have to notice, that users generally lost the ability to see the length of the previous maximum streak badge[4]. In the months after the change, the share of users with a new record is permanently lower compared to before, suggesting that users cared about the maximum streak badge and used previous bests as reference points while streaking (H3).

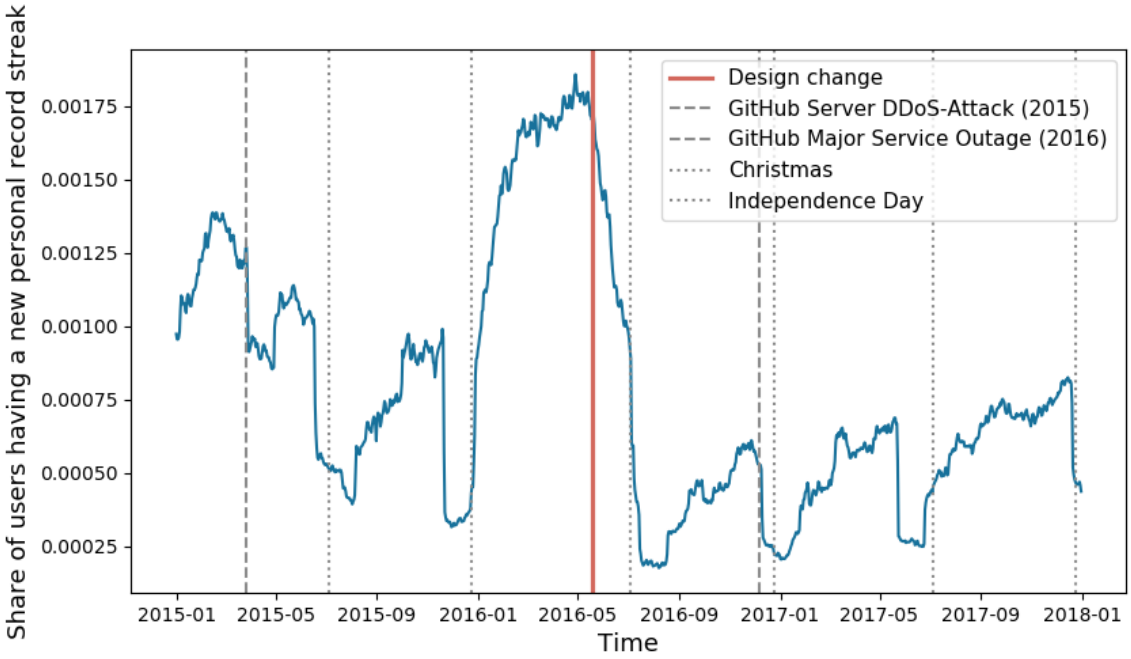

Figure 3.12: *Number of active personal record streaks with a minimum length of 15 in 2015 and 2016. Number of record streaks drops after the design change below values of 2015.*

3.2.2 Lifetime maximum streak

A decrease in new maximum streaks after the change suggests that there could be consequences for the highest streak in a user's lifetime. Thus, we collected the highest streak of each user for 3 years before and after the design change. For the time period afterwards we are ignoring previous bests, which were achieved before the design change happened. To focus on streaks with significant length, we discard users with a highest streak length below 20 days. 5.1% of all male users achieved a maximum streak above this threshold before the design change, while the share drops to 3.8% afterwards. For female users we observe lower shares in general with 3.4% before and 3.1% after the change indicating a more mild decrease. Thus, a lower share of streaking women

[4]See 3.3.2 for a minority of users who still could see the feature

supports the assumption that men are more likely to respond to gamified elements (H4). Moreover the lower decrease in women users after the change suggests that they were less influenced by the design change (H4).

Figure 3.13 is the cumulative distribution of highest achieved streaks separated by gender until $x = 110$. From 2013 to 2016 we observe for each streak length x a higher share of men than women. After the design change, between 2016 and 2019, both distributions drop by roughly the same factor, while men now have a similar distribution with women before the change. Figure 3.14 shows the further trend of the distributions until $x = 210$ with similar findings. Values for women are now fluctuating strongly, as we only compare 2%-5% of the share of original 3% share of women with a streak longer than 20.

In general we observe a lower share of women with a maximum streak longer than 20, which fits to our hypothesis, that women are less interested in streaking (H4). When focusing on the remaining users, who passed the threshold, we observe lower values after the change for both genders. Moreover, even when we focus only on streaking women, women achieve lower maximum streaks compared to men. This once again supports our fourth hypothesis.

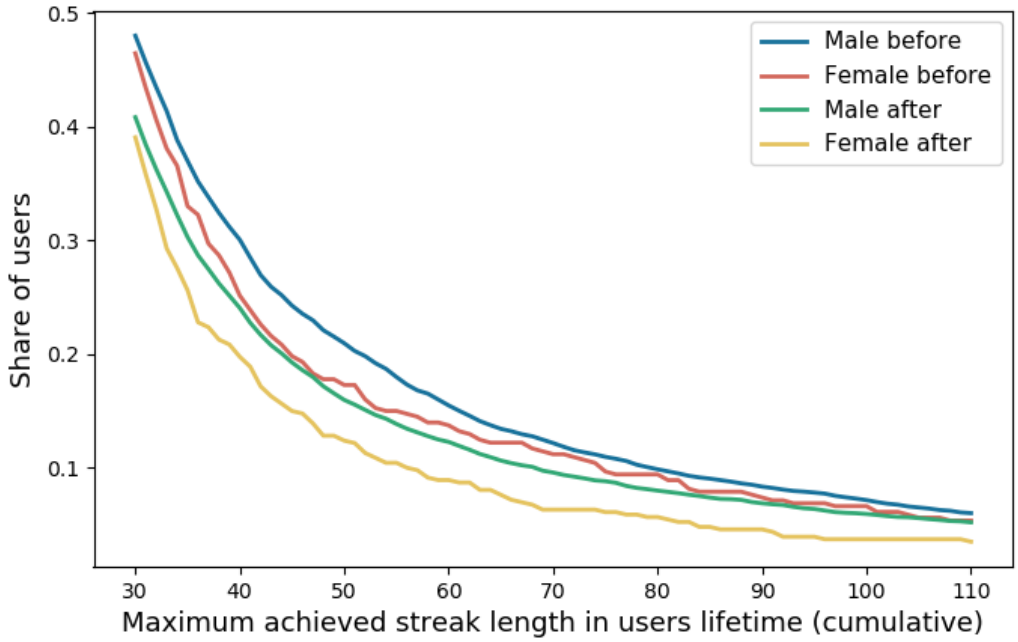

Figure 3.13: *Cumulative distribution of highest achieved streak length ($x \in [30, 110]$) by share of users, considering only users who achieved a minimum streak length of 20. Streaking women achieve lower maximum streaks on average, but are similar affected by the change.*

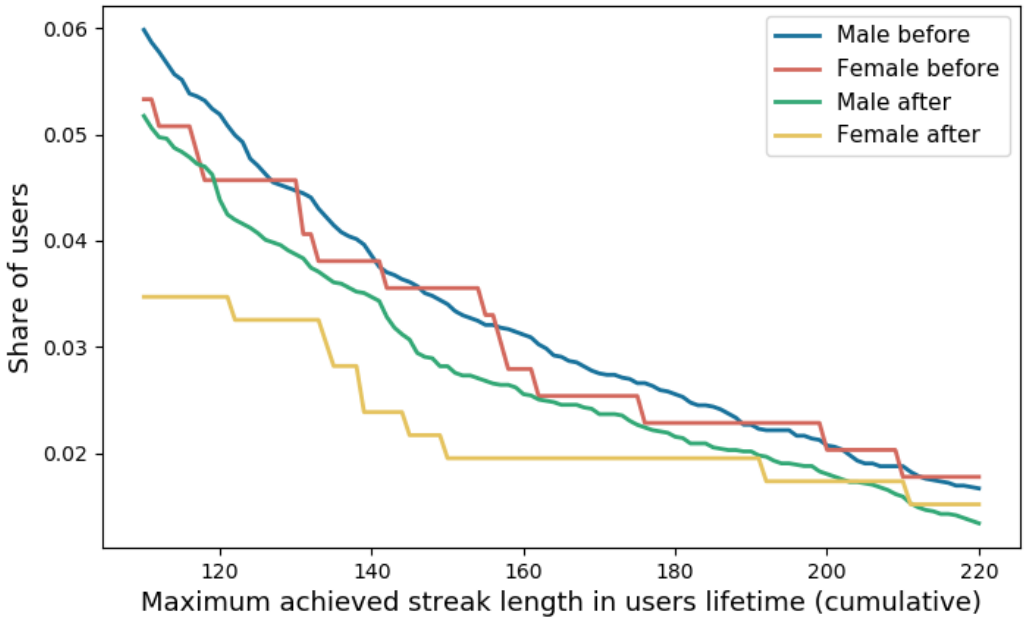

Figure 3.14: *Same cumulative distribution for* $x \in [110, 220]$.

3.3 Incentives for streaking after the design change

In our results so far we have observed a fraction of users who continue streaking after the change. In 3.1.1 and 3.2.2 we saw that even after the change users exist with streak lengths above 200 days. Besides a general unchanged interest in signalling high activity, which is an important factor for popularity on GitHub [53], the remaining contribution graph could still impact users as a gamified element. But there are further individual reasons for continuously streaking. The following part introduces two exemplary groups of users, which both have incentives to continue streaking, even without the streak feature. To conclude, we discuss the issue of cheating users.

3.3.1 Goal based streakers

The streak feature may have been used by users for goal based working or learning. For example a developer learning a new programming language may set the goal to write code in that language every day for one full month. Thus, streaking serves as an intrinsic motivational tool and self-signalling becomes more important than signalling to other users.

As one exemplary goal-based streaking community, we study users participating in the "100DaysOfCode"-challenge[5]. As the name suggests, the challenge's goal is to code at least one hour daily for 100 days in a row. Participants can optionally fork a linked GitHub repository, which serves as a journal template and can be filled with daily individual progress updates. Daily journal updates did not count as a valid contribution for the streak feature, as they are done in a forked repository. But

[5] `https://www.100daysofcode.com/` (February 29, 2020)

we assume that a fraction of users who use the optional template on GitHub, could also use public GitHub projects for their daily coding session.

To collect more users we built a GitHub API scraper to search for further goal based communities on GitHub, which have the same goal of coding 100 days in a row. The resulting projects with journal templates can be found on our GitHub page[6]. For further analysis, we set up a second API scraper, which collects users forking the corresponding template repositories and translates their usernames to IDs in our database. From a collection of 16k forkers of all projects, we found more than 1.6k users in our filtered data (recall that our filtered dataset only contains users for which a location could be inferred). Figure 3.15 shows the daily share of these users between 2015 and 2018, which maintain a streak of length t. In the year before the change, we observe an increase in streaking with several drops (most likely caused by users who reached their goal). But we also observe streaking beyond 150 days. Within the days after the change, a large amount of users stopped streaking immediately for all thresholds t. Users seem to be discouraged by the design change and gave up their goal. However, we observe surviving and new streaks longer than 50 days after the design change in 2017, but with less users participating compared to the year before. The share of users streaking above their goal over 150 days decreased permanently. But do remaining streaking users still streak because of their 100 days goal?

Figure 3.16 represents the total number of users achieving (placebo) goal $g \in \{50, 100, 105, 155\}$ over time (if a user achieves a goal a second time, he will not be counted). The design change seems to have a low impact on these statistics, as many user still achieve significant streak lengths after the change. Eventually the growth of a 50 days achiever could have been affected temporary (H7). But when comparing differences between the number of achievers of different goals, we observe nearly the same increasing gap between achievers of the 50 days placebo / 100 days real goal and achievers of 100 days real / 105 days placebo goal. Thus, many users stopped maintaining their streaks right after hitting the 100 days goal, not even reaching a length of 105 days. These results suggest that users still streak because of the goal based challenge after the change, even without having a streak feature. Moreover, the forked journal with daily updates could have helped to keep track of a streak, as many users stop streaking quite exactly after reaching a length of 100 days. But there are even more possibilities to visualize streaks without the presence of the streak feature, which will be explained in the following.

[6] https://GitHub.com/lukasmoldon/GHStreaksThesis/blob/master/api/source_api.json

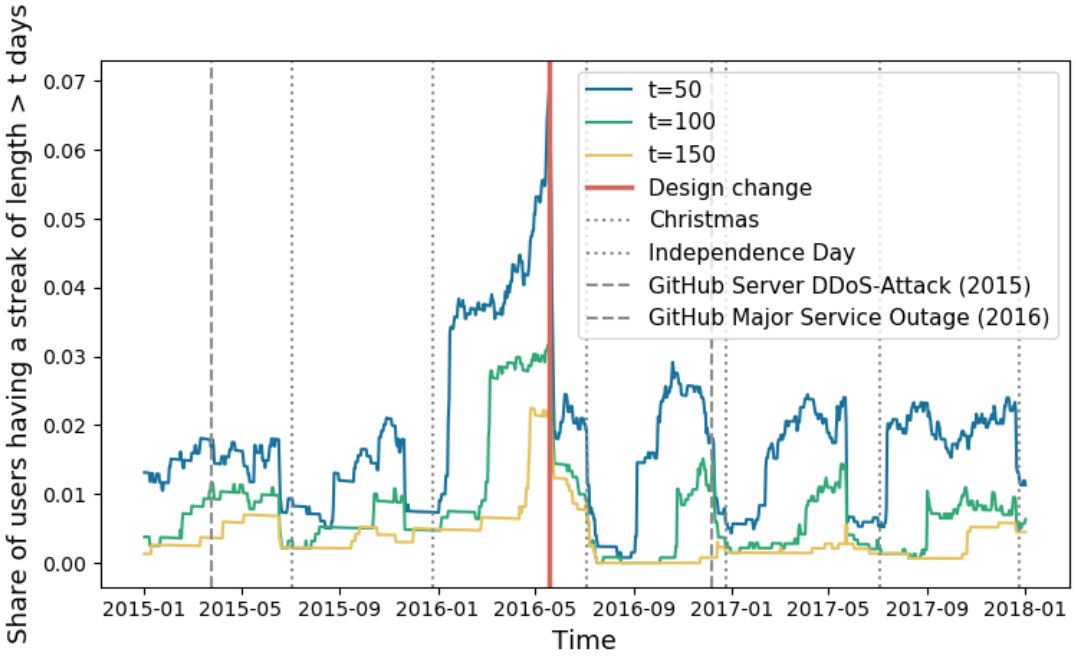

Figure 3.15: *Only considering users, who forked a 100 days goal based GitHub project - Share of users having a streak of length > t days for* $t \in \{50, 100, 150\}$*: Largest drop happened right after removing streaks from GitHub (red line). Share of users having streaks above the 100 days goal decreased after the change permanently.*

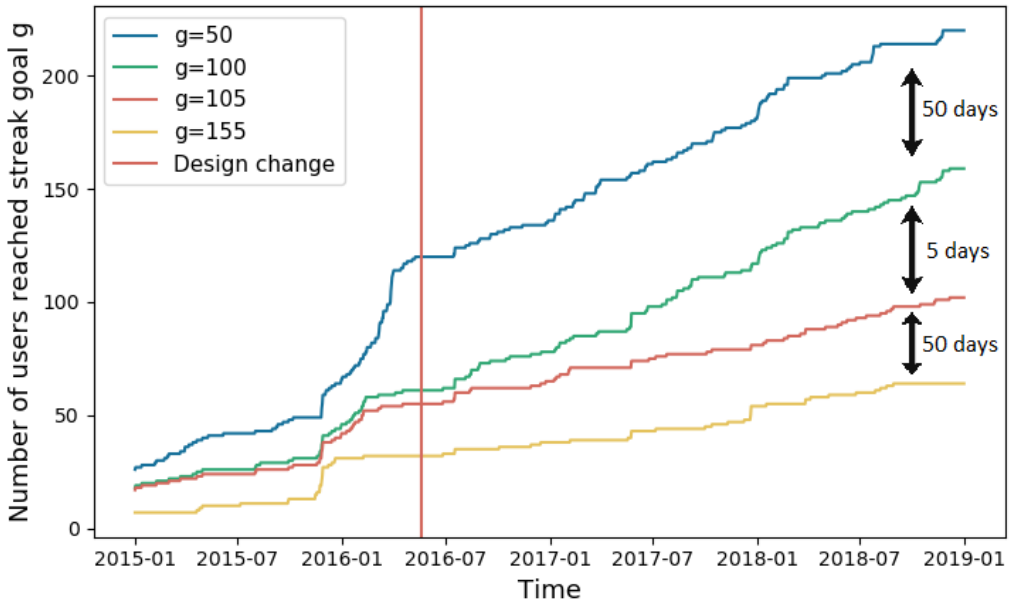

Figure 3.16: *Only considering users, who forked a 100 days goal based GitHub project - Number of users hit the (placebo) goal of g days for* $g \in \{50, 100, 105, 155\}$*: Growth of g=50 achievers decreases after the change, but users still hit new goals. The large difference between achievers of g=100 and g=105 emphasizes the importance to the connection to the 100 days goal.*

3.3.2 Web plugin streakers

As we already described in the introduction, some streakers were angry about the design change and wanted to keep their hard-earned streaks visible. Within several days of the change users started to create web browser plugins which recreated the streak feature for the user's local machine. With such plugins personal streaks remained visible for the owner, but the ability of the user to signal is limited to other plugin users. We collected all subscribers ("stargazers") of one of the most followed recreation projects on GitHub[7] with our API scraper and found 73 users out of 170 in our database (note that subscribing is not required for using the plugin and there may exist more than 170 plugin users).

Due to the small amount of collected users, we could not analyze quantitatively the effect of the design change on streaking. But besides the fact that users put effort into the project and recreated the feature for the community, we observe, similar to the previous analysis of goal-based streakers, new streaks with a length longer than 50 days in 2017. 15% (or 45 in total) of all streaks with a minimum length of 50 days became longer than 150 days after the change, suggesting that web plugins could be incentives to continue streaking.

3.3.3 Cheating streakers

Whenever a user sends code commits from a local machine to the GitHub servers, additional metadata gets sent in a header. It is possible to modify the header without much effort or additional required software, using Git's commit function arguments[8]. Thus, streaks can be a result of contributions with modified timestamps. An extreme example for this behavior is a streak with a length of 32062 days, which "started" 60 years before the internet was invented[9]. Moreover, there are applications, which automatically create a commit each day, resulting in a never ending streak[10].

In our data extraction we filtered out all contributions with invalid timestamps (e.g. before 2008 / after June 2019) and separated users with a large amount of those timestamps. We found no evidence that this behavior became less after the design change. One reason for continuous cheating of streaks could be the popular creation of "graph graffitis" [11], which creates images within the contribution graph with a large amount of fake commits.

As mentioned in 2.1, bots make up a significant number of contributions on GitHub and we thus want to exclude them from our database. Latest research by Tapajit Dey et al. resulted in over 400 detected bots on GitHub [14]. The paper was published at the time of the end of this work, so we could not use their findings to improve our filtering. However, we compared our database

[7] `https://GitHub.com/Naramsim/GitHubOriginalStreak` (Feburary 29, 2020)
[8] `https://stackoverflow.com/questions/3895453/`, (February 29, 2020)
[9] `https://stackoverflow.com/questions/20099235/`, (February 29, 2020)
[10] `https://GitHub.com/theshteves/commit-bot` (February 28, 2020)
[11] `https://GitHub.com/gelstudios/gitfiti` (February 29, 2020)

with their list of bots: We found 177 detected bots[12] in the GHTorrent data dump and already discarded 162 (91.5%) of them through our restrictions in 2.1. Only 9 of the remaining 15 bots in our database have a streak being longer than 7 days.

3.4 Further effects

In the previous sections we studied how streaking changed due to the design changed. We now turn our attention to the hypotheses that the design change had further indirect effects on behavior, like a decrease in weekend work (H8) and consequences for the social network of streaking users on GitHub (H9). In the following we will study these assumptions.

3.4.1 Weekend activity

Users have to commit at inconvenient times like weekends to maintain longer ongoing streaks. Thus, we assume that the overall ratio of contributions on weekends decreased right after the design change (H8). Figure 3.17 shows the weekly ratio of contributions around the change (average of all users) with the mean ratio before and after. We repeated the mean calculation for further time intervals, the mean ratio before is always higher compared to after the change. Shorter time intervals around the change result in higher differences in means. Table 3.2 facilitates a broader view and shows the average and total amount of contributions on weekends for the year before (B) and after (A) the design change. While we observe a drop of 0.09% for all users, this difference increases when considering only users who achieved a minimum streak length of 20 (MIN20) or 30 (MIN30) days in the respective time interval. Here we observe drops between 0.28% and 0.34%. The total amount of contributions on weekends increased for all users, since the GitHub is a growing platform with an increasing number of active users per day. However, the total amount of weekend contributions by streaking users decreased by several million as a result of a fall in streaking users after the design change. To observe the statistical significance of the change, we build a statistical model.

	ALL		**MIN 20**		**MIN 30**	
	B	**A**	**B**	**A**	**B**	**A**
Share of contributions on weekends	0.2188	0.2179	0.2433	0.2399	0.2487	0.2459
Total amount on weekends (in million)	45.3	48.6	13.5	10.3	9.5	7.6

Table 3.2: *Share and total amount of weekend activity for all users and only streaking users (achieving a streak of length 20 or 30 in the respective time interval) for the year before (B) and after (A) the change: Share of weekend work decreases especially for streaking users and total numbers show additionally a decreased number of contributions by less streakers.*

In the following we focus on active users with at least 30 contributions in the respective time interval. To test for statistical significance of the design change on weekend work, we apply the

[12]results: `https://GitHub.com/lukasmoldon/GHStreaksThesis/blob/master/api/bots.json`

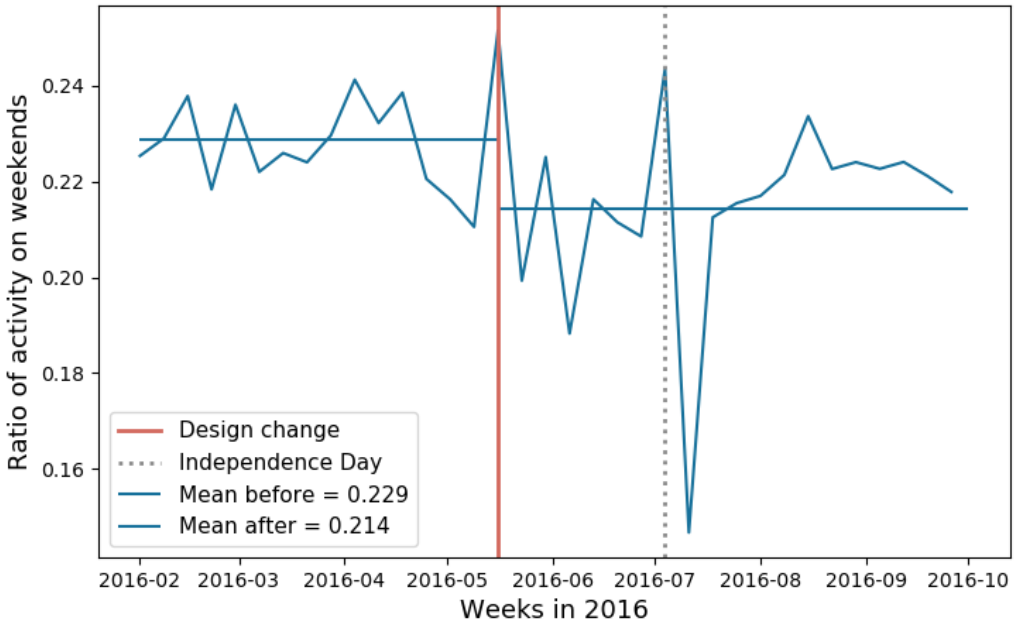

Figure 3.17: *Weekly ratio of contributions on weekends in 2016 for all users: Weekend activity drops after the design change and the Independence Day.*

regression discontinuity design method [28]. Our goal is to fit a linear model on the share of weekend activity per user over time, which estimates the effect of the design change with a treated variable and coefficient. The corresponding linear function is

$$y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot T$$

where y denotes the ratio of weekend activity for a user in week x and T represents the treated variable, with $T = 0$ if x is before the change, $T = 1$ otherwise. We fit our model to the data using the python module RDD [13]. The module estimates the coefficients β_1, β_2 and the intercept β_0 while minimizing $\sum_{i=1}^{n}(y_i - f(x_i))^2$ (the sum of the squared residuals for each observed data point (y_i, x_i)).

Results are shown for different user subgroups in table 3.3. The bandwidth column represents the observed time in weeks (half of the bandwidth directly before/after the change). We are not observing the week of the design change itself, to avoid mixing weekdays before the change with a treated weekend directly afterwards. A larger number of observations (or higher bandwidth) reduces β_1 to nearly 0, indicating time (without treatment) being an insignificant factor for the weekend activity in general.

[13] `https://GitHub.com/evan-magnusson/rdd` (February 23, 2020)

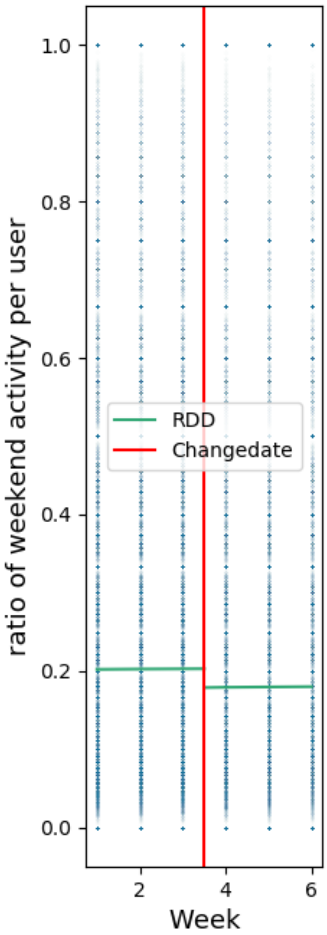

Figure 3.18: *Plot of the weekend RDD*

In contrast, nearly all treated coefficients are negative and we observe $|\beta_2| > 0.02$ for nearly all tests supporting H8. Only for women and Chinese users we observe lower or positive treatment, which is related to our fourth and sixth hypothesis, that women are less influenced by gamification elements and that the user's origin matters. Unfortunately both results including additionally Germany are not significant at $p \gg 0.05$. In contrast to all users, we find higher treated coefficients for men and users from the UK or USA, with values up to -0.0667 and significant p values under 1%. Figure 3.18 is a plot of the regression function for all users with bandwidth 6. Blue dots represent the share of weekend activity of a single user. In the top 10% area more blue dots are present before than after the change.

subgroup	**bandwidth**	**#obs.**	β_0 **(intercept)**	β_1 **(x coeff.)**	β_2 **(treated coeff.)**	**p value**
ALL	2	73433	0.0582	0.0358	-0.0985	<0.001
ALL	4	144726	0.1843	0.0050	-0.0365	<0.001
ALL	6	214249	0.2016	0.0004	-0.0241	<0.001
MALE	4	77395	0.1728	0.0062	-0.0392	<0.001
MALE	6	114719	0.1985	-0.0006	-0.0209	<0.001
FEMALE	4	4618	0.1482	0.0076	-0.0347	0.172
FEMALE	6	6864	0.1951	-0.0060	0.0005	0.976
USA	4	46757	0.1502	0.0097	-0.0473	<0.001
CHINA	4	9740	0.2611	-0.0114	0.0043	0.806
UK	4	7920	0.1456	0.0136	-0.0677	<0.001
GERMANY	4	7890	0.2073	0.0005	-0.0211	0.299

Table 3.3: *Results of the RDD weekend model for different subgroups. A larger number of observations (or higher bandwidth) reduces β_1 to nearly 0. $|\beta_2| > 0.02$ for all tests except women and China. For female, Chinese and German users we could not achieve a significant p value.*

In order to test the robustness of our findings, we carried out a series of placebo tests using the same model with the design change artificially set to different dates in 2016. To keep results comparable, we again only focus on active users with at least 30 contributions in the respective time interval and use bandwidth 4. Figure 3.19 shows the resulting treated coefficients for all tests with the placebo date in week x. Before the change, we observe no higher treated coefficient than the original one of -0.0365 and larger 2.5% confidence intervals in general. After the change we observe fluctuating coefficients around the Independence Day but also similar values in September and October. We make two points here: first, the fall in weekend work around the fourth of July weekend is clearly compensated by overwork on neighboring weekends. Second, all placebo points before the design change show no difference as large as the real design change. Yet we must concede that our model only hints at a changed behavior and does not prove beyond a doubt that the change decreased weekend work. We continue this discussion in the future work section.

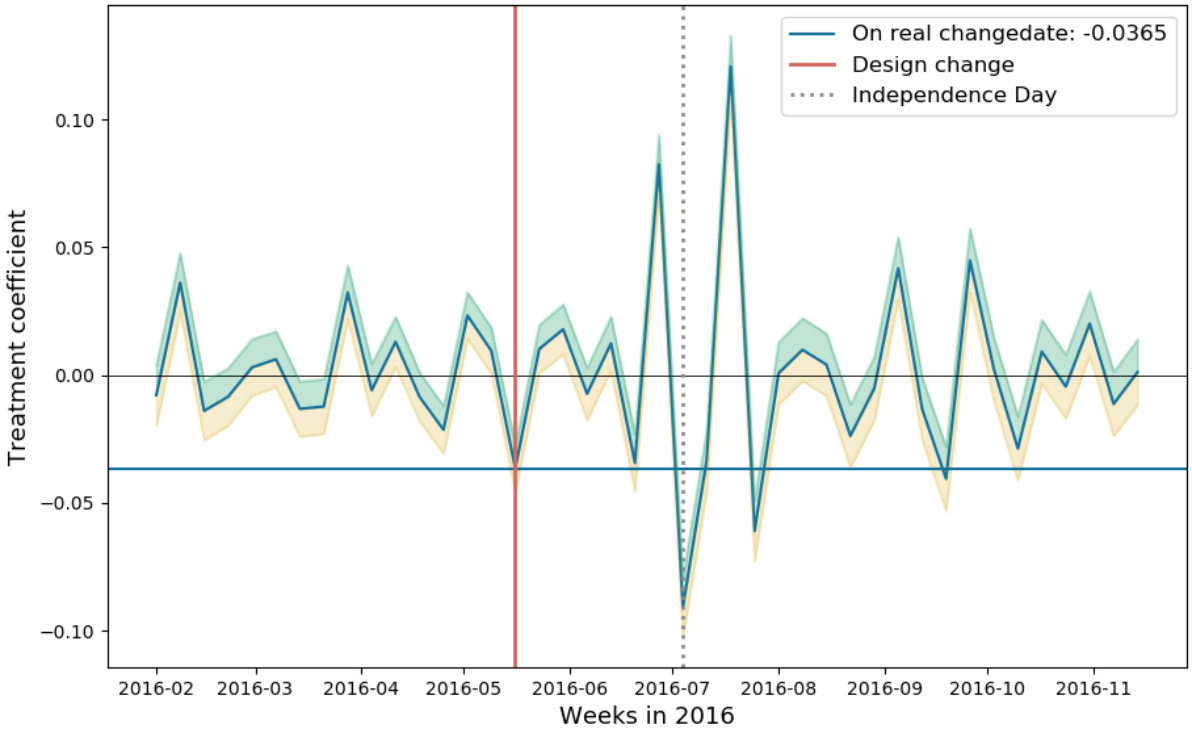

Figure 3.19: *Results of the RDD weekend placebo test with fake change dates and bandwidth 4.*

3.4.2 Social network

Signalling long streaks requires other users to notice a user's personal achievement. Thus, we assume that streaking users tend to be connected on GitHub, and that the design change muted the correlations in their streaking behavior (H9). To analyze this hypothesis, we generated a social network from the follower data of GHTorrent. Users can use the functionality to stay in contact with users the have followed and to receive activity updates about them in their news feed. GHTorrent's data is limited to follower connections which still existed in June 2019 but contains

the creation date of a follower connection. So it is possible to create networks for different days, while we can only add new existing connections, which remain until 2019, over time. However, an GitHub API sample test of 1,000 users found that only 2% of all follows between May 2016 and June 2019 were deleted before June 2019.

Nodes represent users in the network, while links between them are mutual follower connections. Since we want to focus on strong ties, we do not observe single/non-mutual follows and discarded all nodes with a degree of 0. The resulting undirected network on May 18 in 2016 (before the change) contains 146k nodes and 253k links with an average degree of $< k >= 3.46$ and a maximum degree of $k_{max} = 2343$. The network is divided into 11k components, while the largest component contains more than 81% of all nodes.

To introduce streaking into the network each node received an attribute of "streaker" or "no streaker" depending on the value of the personal maximum streak badge right before the change, for different thresholds $t \in \{8, 15, 32\}$. If the maximum streak length of a user is at least t days long, the user is considered to be a streaker in test t. Now we analyze the tendency of users to be connected with other users that share the same attribute. Thus, we calculate the attribute assortativity coefficient by Newman (Eq. (2) in [39]) using NetworkX [14]. Generally, assortativity r is in $[r_{min}, 1]$, where $r = 1$ means the network is perfectly assortative, $r = 0$ indicates a random distribution of the attribute and $r = r_{min}$ stands for perfect disassortativity. There exists a network specific $r_{min} \in [-1, 0)$, as perfect disassortativity is normally closer to a randomly mixed network than to perfect assortativity (Eq. (3) in [39]), which is not relevant for our analysis of positive assortativity. To analyze statistical significance, we repeat all tests for 1,000 copies of the original network, in which the streaking label is randomly assigned.

(2016/05/18)	*real network*		*random attr. networks (avg)*		
threshold t	**assortativty**	**P(SN | S)**	**assortativty**	**P(SN | S)**	**z-score (assortativity)**
8	0.0886	0.3861	-0.0001	0.1787	41.6
15	0.0791	0.3402	-0.0001	0.0821	36.9
32	0.0448	0.2532	-0.0001	0.0271	21.2

Table 3.4: *Network assortativity and P(SN | S) = P(n has streaking neighbor | n is streaker) for the real network on 2016/05/18 and 1k copies with randomly drawn attributes from the origin distribution. Both values differ significantly from the random experiment, indicating a connection between streaking and the network structure.*

Table 3.4 shows that the streaker attribute is not randomly distributed, as we observe an assortativity around 0 for the randomized networks and between 0.04 and 0.09 for the real network. We calculated a z-score to test the statistical significance of the difference in assortativity between the empirical graph and the random simulated graphs, with $p \gg 1.96$ for all tests. Values decrease

[14] `https://networkx.GitHub.io/documentation/stable/reference/algorithms/assortativity.html` (February 25, 2020)

with an increasing streaker threshold t, since there exist less streakers and remaining streaking nodes have a higher fraction of non-streaking neighbors. To emphasize the difference, we computed P(SN | S) = P(node n has streaking neighbor | node n is streaker). Here we see a significant difference: The probability that a streaker is connected to some other streaker is 38.6% compared to the random distributed attribute with 17.7% for $t = 8$. With an increasing t the difference between the networks increases too. At $t = 32$ we observe P(SN | S) = 25.3% compared to 2.8% as average in the random networks.

We already know from our lifetime maximum streak analysis, that it is quite rare to observe streak records longer than a month. This is the reason why we observe such a low probability that two users with a long record streak are randomly (and mutually) connected. But in our real network, every fourth high streaker is connected with another high streaker. So streaking users are not only connected among themselves in general, they are also commonly connected to other streakers with a similar level of streaking records.

(2017/05/20)	*real network*		*random attr. networks (avg)*		
threshold t	**assortativty**	**P(SN | S)**	**assortativty**	**P(SN | S)**	**z-score (assortativity)**
8	0.0911	0.2653	-0.0001	0.1402	37.7
15	0.0554	0.2031	0.0001	0.0471	22.6
32	0.0211	0.1118	-0.0001	0.0106	8.3

Table 3.5: *Same calculations for the real network on 2017/05/20. The difference from the random network is still significant but decreased for P(SN | S).*

This analysis of assortativity of users according to their tendency to streak is an example of a well-studied phenomenon in networks: homophily [36], the tendency to observe similar individuals together in some contexts. Network scientists have long studied two broad mechanisms that can explain observed homophily: sorting, by which individuals with similar tastes and preferences are more likely to become friends, and influence, by which individuals become more like their friends by adopting their behavior. In many cases, and likely in our situation, both factors are present. Highly active users who like to signal their performance are likely friends with similar users. Such users may be driven to even greater levels of streaking by the desire to imitate or even compete with their network neighbors.

In general, these two factors are confounded: it is not possible to distinguish between them using observational data [45]. However, with experiments or quasi-experiments it can be possible to distinguish the two [4]. If we repeat the analysis of homophily among streakers after the change, we can test if both sorting and influence are at play. If only sorting is at play, there should be no change in the observed assortativity levels. If only influence is present, then there should be little or no assortativity remaining. Results in between suggest that both effects were present before, and that the design change blocked an important channel for influence. Indeed this is what we hypothesize: that some users were driven to extend their streaks because they observed higher

totals among their neighbors in the network, and that the design change ended this phenomenon.

We thus created the same network one year after the change and only observed streak records in these 12 months for the streaker attribute. Besides 20k new existing nodes, there is an overall increase of 13.4% in edges and a general increase of connectivity among nodes. Table 3.5 shows that the network is assortative and that the remaining streakers are still connected. But overall values decreased compared to the random networks and table 3.4. Decreased z-scores represent a less deviating assortativity from the random simulation. For $t = 8$ only every fourth user is connected to another streaker, while one year before we observed a chance of 38,6%. So the design change did not affect the network structure itself, but influenced the tendency of streaking users to connect. Now, remaining streakers are much less connected among themselves (H9) and only every tenth user with a streak length longer than 32 is mutual connected with another such user (compared to 25% before). This suggests that the signals provided by the streak counters were indeed a conduit for peer effects in the social network of GitHub developers.

4

Conclusion

4.1 Summary of findings

The design change of GitHub is a convincing natural experiment - an unanticipated external shock that facilitates causal interpretation of resulting changes in behavior. From one day to the next, thousands of GitHub users went from seeing their current and all-time longest streaks every time they logged onto the platform, to having no straightforward way to observe these counts. The results presented in this thesis confirm that the GitHub site design had significant effects on behavior in multiple ways. Specifically, we found statistical evidence that streaks became shorter (H1) and that extreme streaks became more rare after the design change (H2). A higher share of days with only one contribution at the end of a streak before the change suggests that before the change users tried to maintain streaks beyond their normal coding activities, supported by an non-uniform distribution of contribution over streaks in general. Additionally we observed different behavior for women and men (H4), as men tended to be more interested in streaking in general. Even when only focusing on streaking users, men achieved longer maximum streaks on average before and with decreased values (for men and women) after the change. The removal of the maximum streak counter seems to have caused a decreased interest in beating previous record streaks in the long term (H3).

We found further differences between different user groups. Recurring 5 and 12 day streaks starting on Mondays, most likely driven by employees using GitHub at work, resisted the design change (H5), but longer streaks became more rare. We also observe regional differences, as users from China kept on streaking on a slightly less intense but similar level compared to western countries, where streaking rates dropped significantly (H6).

We also explored reasons to continue streaking after the change. Besides users who recreated the feature on their local machines, various goal-based communities used journal templates to keep track of their streaking goal. Here we observed streaking as an intrinsic motivational tool for

personal progress, which is in contrast to streaking for signalling achievements to others. Thus, different motivations led temporarily to similar behavior. Goal based streakers continued following their goals, but had a decreased interest in maintaining streaks beyond that goal after the change (H7).

We also applied more sophisticated methods to measure the magnitude and significance of behavioral changes besides streaking itself. These methods include a regression discontinuity design on weekend activity, which hints at a decrease of weekend work (H8), when users lost the ability to signal personal streaks. Moreover we found consequences for the social follower network on GitHub (H9), as streaking behavior was significantly more clustered before the change than afterwards, suggesting that some significant part of pre-change streaking behavior was driven by social influence. It seems that prior to the change some users observed streaks of their neighbors and increased their activity to match them.

4.2 Discussion

The thesis presented specific findings on the impact of signaling opportunities on developer behavior, and the importance of site design in general. The use of gamified counters motivated users to contribute to GitHub daily for long time periods. This increased the overall participation and number of daily active users over time. But contributing to GitHub for many days successively raises the risk of unwanted side effects, especially for the open source community consisting of many developers, who contribute voluntarily in their free-time next to a full-time job [10]. Our results suggest that users worked more on the weekend when the streak feature was enabled, while medical studies emphasize the importance of weekend recovery for both personal health and also job performance [5].

When GitHub introduced streaks together with the contribution calendar, they advertised it as a meaningful community feature which summarizes the user's activity [6]. However, the new design incentivized working many days in a row without a break. Over the years users streaked with increasing total lengths, especially in 2016 (see 3.1.1), with several users having streak lengths over 1000 days[1]. The community even provided tools like desktop menu bar extensions[2] or automatic email notifications[3] to remind users about their daily contribution. Users also exploited the feature using fake commit dates. As a result some users had a corrupted profile page and contacted the official support for help[4]. Besides additional work for GitHub's technical support, this abuse of the streak feature devalued honest long streaks. Some combination of these reasons likely pushed

[1] `https://www.freecodecamp.org/news/GitHub-broke-my-1-000-day-streak-6ec0c4c3a7d9/`, (March 13, 2020)
[2] `https://GitHub.com/jamieweavis/streaker` (March 13, 2020)
[3] `https://GitHub.com/motdotla/GitHub-streaker` (March 13, 2020)
[4] `https://GitHub.com/isaacs/GitHub/issues/370` (March 13, 2020)

GitHub to implement the design change. In the official announcement GitHub explained their decision as a shift to help users focus on the quality of work and not on the duration of working [7].

Gamification has risks but also benefits, depending on the implementation of the corresponding gamified element. In our example the on-going streak counter "punished" a single offline day. The maximum streak counter encouraged very long periods of uninterrupted contributions. Such thresholds may increase the excitement for specific achievements, but may also increase stress or pressure.

To draw a contrast, on Stack Overflow, another important platform for OSS developers, we also observe the use of badges, but there activity counters do not decrease or reset themselves over time. Besides the aspect of time, GitHub's streak feature introduced a special type of badge. While Anderson et al. describe optimal thresholds for continuous values (like counters) where single badges should be awarded [3], GitHub's streak feature did not provide such fixed goals or boundaries. This could increase the risk of (endless) unhealthy behavior and over-commitment. Furthermore a focus on streaking could reduce the attention for the work itself. This leads us to the open task to measure the quality of contributions to analyze whether the drive to signal can have negative consequence, which will be addressed in the future research section.

We do not completely reject gamification on platforms like GitHub, but suggest that gamification should be employed in a way to decrease the risk of creating harmful incentives and negative side effects. Our research showed that only a minority of users made use of the streak feature, but seemed to be strongly motivated by it. A potential reason could be that streaking over weeks differs significantly from common working habits and thus it is less attractive for the vast majority of users to participate. If badges would reward quality instead of quantity over time, it is imaginable that more users could be reached. Github has many kinds of activities which may be deserving of gamification. For instance, stars and forks on projects give an idea of the usability and quality of a projects. Moreover, cheating in this context would be harder (but still possible) as these functionalities require actions from other users. This could increase the overall value of badges and increase the interest of providing useful code. Badges summarising these type of feedback actions could also function as better credentialing system, as pointed out by Anderson et al. [3]. In the specific context of open-source software, badges could also be employed to steer users towards underserved activities such as the creation of documentation and docstrings, resolving issues, and answering questions from users. Encouraging social behavior, such as leaving comments or accepting merges could also be a noble goal for future badges.

Besides encouraging individual progress, gamification can motivate collaborative working on online platforms like GitHub. Previous case studies indicate that gamification can increase collaboration [31] and knowledge contribution [46] within groups. However, commonly used online col-

laborative working platforms do not utilize gamified elements or focus only on rewarding quantity instead of quality [37]. Our results indicate that integrating gamified elements into the platform design can strongly motivate platform users to perform specific actions, which emphasizes the potential of steering user behavior. Moreover, we found consequences for the social platform network in 3.4.2, as streaking users were stronger mutually connected with a present streak feature. Thus, implementing gamification on all sorts of collaborative working platforms could facilitate the deployment of personal abilities and knowledge within groups and could have a positive impact on the social platform network. As an example, crowd-working platforms divide work between many (voluntary) users and allow to achieve major results collaboratively. Currently this model is growing in popularity, especially the case of tasks which are outsourced by companies and offered to a large pool of competing crowd workers [30]. Research shows, that gamification can increase the worker's motivation besides monetary incentives and improve the quality of work on these platforms [17], with important implications for the labor market. This underlines the importance of studying gamification in all kinds of applications.

In general one should consider the trade-off between benefits and risks when implementing gamified elements. Gamification can support the joy of solving difficult tasks like coding (ideally in a healthy way), illustrate an individual's personal development, facilitate collaborative work on online platforms and motivate participation in OSS. But we also have to recognize that not everyone likes gamified elements at work, while others could focus more on the game than on the serious working context [19].

4.3 Limitations

We note several limitations that may be addressed in subsequent studies. The streak computation is very sensitive to small changes in the source data, already a single missing day in our data would end all streaks immediately. Fortunately we did encounter such incidents. But our underlying database, which we have computed from the GHTorrent data dump, contains minor inconsistencies as described in 2.2. These small errors could possibly end streaks and we observed several times inexplicable local but significant changes in values. In 3.3.3 we recognized technical possibilities to edit header information and modify contribution times. We deleted users with obvious irregularities, but we potentially still have remaining "cheating" users in our observed user group.

Moreover, we had to discard users without available location information from our research, since streaks were evaluated in the user's local time zone but data is saved uniformly in UTC-0. This introduces an unstudied selection bias, as users providing such advanced and optional information could be more professional and thus not representative for the whole developer community on GitHub. The same issue appears when analysing behavior associated with differences in gender, since we only obtained the gender of users who shared their full name on the platform. Addition-

ally we used a simple location based gender detection algorithm, which potentially classified the gender of some users incorrectly and does not cover every region in the world.

When discussing incentives for streaking we must acknowledge that the remaining contribution graph could still motivate streaking as gamified element. Moreover we discussed technical possibilities to recreate the original streak feature in 3.3.2. Furthermore, streaking or generally behavior on online platforms depends on a countless amount of (individual) factors. We have seen holidays and service downtimes affecting streaking rates, and it is imaginable that many other aspects are influencing our statistics. To conclude, it was difficult to identify streaking users motivated by the feature, as streaks can happen unintentionally as described in 3.1.2.

4.4 Future research

The results of this thesis demonstrate that the sudden removal of gamified elements had a significant impact on user behavior for at least a part of GitHub's developer community. Significant changes in the volume, consistency, and timing of contributions likely have consequences for the output. This leads us to the unanswered question of whether user code improves when users focus on their projects rather than the potential to signal activity. In general, measuring code quality is a difficult task with subjective aspects. Past work measured code quality on GitHub by comparing the ratio of accepted pull requests of different groups [47] and it could be worth to investigate this question before and after the design change. Also our investigations about a decrease in weekend work merit a closer look. For instance, our linear regression discontinuity model is basic and could be expanded by including further influencing factors or by focusing on a subset of users with specific characteristics.

Direct contributions are not the only fundamental activity on GitHub, and changes in social behavior among streakers is still understudied. We made some preliminary investigation into the number of comments and the sentiment of those comments made by (streaking) users before and after the change. Unfortunately, only commit and pull-request comments are part of the GHTorrent data, while valuable comments on issues (frequently used for giving feedback to a developer) are only available through the restricted GitHub API. We did not observe significant changes in commit and pull-request comments for the available data, but we suggest that more specific kinds of behavior changed. The future study of those activities which were *not* recognized by the gamified feature is especially promising, as we expect a substitution from rewarded activities to unrewarded activities across the design change. Our research on days with one single contribution representing the minimum effort required for maintaining streaks suggests that the relative frequency of unrewarded platform actions increased. For example studying the usage of functionalities like forking, giving stars, watching repositories, commenting and following could deliver new knowledge on how gamification influences unrewarded actions and if there was an increased tendency of usage

after the design change.

To conclude, we assume that the diversity of contributions could be affected by the design change. In our results we observed a decreased tendency of single contribution days appearing at the end of a streak's lifetime after the change, which hints at previous exploiting behavior to keep streaks alive. When examining the trade-off between exploration and exploitation, future research could study the amount of projects each user is contributing to or the diversity of programming languages employed.

Bibliography

[1] Kristen M Altenburger et al. “Are there gender differences in professional self-promotion? an empirical case study of linkedin profiles among recent mba graduates”. In: *Eleventh International AAAI Conference on Web and Social Media*. 2017.

[2] Ashton Anderson and Etan A Green. “Personal bests as reference points”. In: *Proceedings of the National Academy of Sciences* 115.8 (2018), pp. 1772–1776.

[3] Ashton Anderson et al. “Steering user behavior with badges”. In: *Proceedings of the 22nd international conference on World Wide Web*. ACM. 2013, pp. 95–106.

[4] Sinan Aral and Christos Nicolaides. “Exercise contagion in a global social network”. In: *Nature communications* 8.1 (2017), pp. 1–8.

[5] Carmen Binnewies, Sabine Sonnentag, and Eva J Mojza. “Recovery during the weekend and fluctuations in weekly job performance: A week-level study examining intra-individual relationships”. In: *Journal of Occupational and Organizational Psychology* 83.2 (2010), pp. 419–441.

[6] The GitHub Blog. *Introducing Contributions*. `https://github.blog/2013-01-07-introducing-contributions/`. Accessed: 02.03.2020.

[7] The GitHub Blog. *More contributions on your profile*. `https://github.blog/2016-05-19-more-contributions-on-your-profile/`. Accessed: 29.02.2020.

[8] Amiangshu Bosu and Kazi Zakia Sultana. “Diversity and Inclusion in Open Source Software (OSS) Projects: Where Do We Stand?” In: *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE. 2019, pp. 1–11.

[9] Samridhi Shree Choudhary et al. “Modeling Coordination and Productivity in Open-Source GitHub Projects”. In: (2018).

[10] Maëlick Claes et al. “Do programmers work at night or during the weekend?” In: *Proceedings of the 40th International Conference on Software Engineering*. 2018, pp. 705–715.

[11] Laura Dabbish et al. “Social coding in GitHub: transparency and collaboration in an open software repository”. In: *Proceedings of the ACM 2012 conference on computer supported cooperative work*. ACM. 2012, pp. 1277–1286.

[12] Paul A David and Joseph S Shapiro. “Community-based production of open-source software: What do we know about the developers who participate?” In: *Information Economics and Policy* 20.4 (2008), pp. 364–398.

[13] Sebastian Deterding et al. “Gamification. using game-design elements in non-gaming contexts”. In: *CHI’11 extended abstracts on human factors in computing systems*. ACM. 2011, pp. 2425–2428.

[14] Tapajit Dey et al. “Detecting and Characterizing Bots that Commit Code”. In: *arXiv preprint arXiv:2003.03172* (2020).

[15] Nadia Eghbal. “Roads and Bridges”. In: *The Unseen labor behind our digital infrastructure* (2016).

[16] Sebastian von Engelhardt, Andreas Freytag, and Christoph Schulz. “On the geographic allocation of open source software activities”. In: *International Journal of Innovation in the Digital Economy (IJIDE)* 4.2 (2013), pp. 25–39.

[17] Oluwaseyi Feyisetan et al. “Improving paid microtasks through gamification and adaptive furtherance incentives”. In: *Proceedings of the 24th International Conference on World Wide Web*. 2015, pp. 333–343.

[18] Denae Ford et al. “Paradise unplugged: Identifying barriers for female participation on stack overflow”. In: *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM. 2016, pp. 846–857.

[19] Matthieu Foucault et al. “Fostering good coding practices through individual feedback and gamification: an industrial case study”. In: *Empirical Software Engineering* 24.6 (2019), pp. 3731–3754.

[20] The Linux Foundation. *Corporate Open Source Programs are on the Rise as Shared Software Development Becomes Mainstream for Businesses*. `https://www.linuxfoundation.org/uncategorized/2018/08/corporate-open-source-programs-are-on-the-rise-as-shared-software-development-becomes-mainstream-for-businesses/`. Accessed: 03.03.2020.

[21] Rishab Aiyer Ghosh. “Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU”. In: (2007).

[22] Jesus M Gonzalez-Barahona et al. “Geographic origin of libre software developers”. In: *Information Economics and Policy* 20.4 (2008), pp. 356–363.

[23] Georgios Gousios and Diomidis Spinellis. “GHTorrent: GitHub’s data from a firehose”. In: *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*. IEEE. 2012, pp. 12–21.

[24] Georgios Gousios et al. “Lean GHTorrent: GitHub data on demand”. In: *Proceedings of the 11th working conference on mining software repositories*. 2014, pp. 384–387.

[25] Scott Grant and Buddy Betts. “Encouraging user behaviour with achievements: an empirical study”. In: *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE. 2013, pp. 65–68.

[26] David Hinds and Ronald M Lee. "Social network structure as a critical success condition for virtual communities". In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. IEEE. 2008, pp. 323–323.

[27] Denise Hruby. *Young Chinese are sick of working long hours*. `https://www.bbc.com/worklife/article/20180508-young-chinese-are-sick-of-working-overtime`. BBC. 2018. Accessed: 10.03.2020.

[28] Guido W Imbens and Thomas Lemieux. "Regression discontinuity designs: A guide to practice". In: *Journal of econometrics* 142.2 (2008), pp. 615–635.

[29] JetBrains. *The State of Developer Ecosystem 2019 - Team Tools*. `https://www.jetbrains.com/lp/devecosystem-2019/team-tools/`. Accessed: 03.03.2020.

[30] Aniket Kittur et al. "The future of crowd work". In: *Proceedings of the 2013 conference on Computer supported cooperative work*. 2013, pp. 1301–1318.

[31] Antti Knutas et al. "Increasing collaborative communications in a programming course with gamification: a case study". In: *Proceedings of the 15th International Conference on Computer Systems and Technologies*. 2014, pp. 370–377.

[32] Karim R Lakhani and Robert G Wolf. "Why hackers do what they do: Understanding motivation and effort in free/open source software projects". In: (2003).

[33] Raymond Zhong Lin Qiqing. *'996' Is China's Version of Hustle Culture. Tech Workers Are Sick of It.* `https://www.nytimes.com/2019/04/29/technology/china-996-jack-ma.html`. The New York Times. 2019. Accessed: 10.03.2020.

[34] Momin M Malik and Jürgen Pfeffer. "Identifying platform effects in social media data". In: *Tenth International AAAI Conference on Web and Social Media*. 2016.

[35] Anna May, Johannes Wachs, and Anikó Hannák. "Gender differences in participation and reward on Stack Overflow". In: *Empirical Software Engineering* (2019), pp. 1–23.

[36] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a feather: Homophily in social networks". In: *Annual review of sociology* 27.1 (2001), pp. 415–444.

[37] Christian Meske et al. "Social collaboration and gamification". In: *Gamification*. Springer, 2017, pp. 93–109.

[38] Courtney Miller et al. "Why Do People Give Up FLOSSing? A Study of Contributor Disengagement in Open Source". In: *IFIP International Conference on Open Source Systems*. Springer. 2019, pp. 116–129.

[39] Mark EJ Newman. "Mixing patterns in networks". In: *Physical Review E* 67.2 (2003), p. 026126.

[40] Cassandra Overney et al. "How to Not Get Rich: An Empirical Study of Donations in Open Source". In: (2020).

[41] Huilian Sophie Qiu et al. “Going farther together: The impact of social capital on sustained participation in open source”. In: *Proceedings of the 41st International Conference on Software Engineering*. IEEE Press. 2019, pp. 688–699.

[42] Ayushi Rastogi et al. “Relationship between geographical location and evaluation of developer contributions in github”. In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2018, pp. 1–8.

[43] Gregorio Robles et al. “FLOSS 2013: A survey dataset about free software contributors: challenges for curating, sharing, and combining”. In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. 2014, pp. 396–399.

[44] Anita Sarma et al. “Hiring in the global stage: Profiles of online contributions”. In: *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*. IEEE. 2016, pp. 1–10.

[45] Cosma Rohilla Shalizi and Andrew C Thomas. “Homophily and contagion are generically confounded in observational social network studies”. In: *Sociological methods & research* 40.2 (2011), pp. 211–239.

[46] Ayoung Suh and Christian Wagner. “How gamification of an enterprise collaboration system increases knowledge contribution: an affordance approach”. In: *Journal of Knowledge Management* (2017).

[47] Josh Terrell et al. “Gender differences and bias in open source: Pull request acceptance of women versus men”. In: *PeerJ Computer Science* 3 (2017), e111.

[48] Asher Trockman et al. “Adding sparkle to social coding: an empirical study of repository badges in the npm ecosystem”. In: *Proceedings of the 40th International Conference on Software Engineering*. ACM. 2018, pp. 511–522.

[49] Jason Tsay, Laura Dabbish, and James Herbsleb. “Influence of social and technical factors for evaluating contribution in GitHub”. In: *Proceedings of the 36th international conference on Software engineering*. ACM. 2014, pp. 356–366.

[50] Matthew Van Antwerp and Greg Madey. “The importance of social network structure in the open source software developer community”. In: *2010 43rd Hawaii International Conference on System Sciences*. IEEE. 2010, pp. 1–10.

[51] Bogdan Vasilescu et al. “Gender and tenure diversity in GitHub teams”. In: *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 2015, pp. 3789–3798.

[52] David A Wheeler. *Why open source software/free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers*. 2007.

[53] Joicymara Xavier, Autran Macedo, and Marcelo de Almeida Maia. “Understanding the popularity of reporters and assignees in the Github.” In: *SEKE*. 2014, pp. 484–489.

[54] Li Xiaotian. “The 996. ICU Movement in China: Changing Employment Relations and Labour Agency in the Tech Industry”. In: *Made in China Journal* (2019).