Poster: Communication in Open-Source Projects–End of the E-mail Era?

Verena Käfer
Daniel Graziotin
Ivan Bogicevic
Stefan Wagner
Jasmin Ramadani
{verena.kaefer|daniel.graziotin|ivan.bogicevic|stefan.wagner|jasmin.ramadani}@informatik.uni-stuttgart.de
Institute for Software Technology, University of Stuttgart
Stuttgart, Germany

ABSTRACT

Communication is essential in software engineering. Especially in distributed open-source teams, communication needs to be supported by channels including mailing lists, forums, issue trackers, and chat systems. Yet, we do not have a clear understanding of which communication channels stakeholders in open-source projects use. In this study, we fill the knowledge gap by investigating a statistically representative sample of 400 GitHub projects. We discover the used communication channels by regular expressions on project data. We show that (1) half of the GitHub projects use observable communication channels; (2) GitHub Issues, e-mail addresses, and the modern chat system Gitter are the most common channels; (3) mailing lists are only in place five and have a lower market share than all modern chat systems combined.

CCS CONCEPTS

• **Information systems → Data mining**; **Internet communications tools**; • **Software and its engineering → Open source model**;

KEYWORDS

communication, open-source, mining software repositories

ACM Reference Format:
Verena Käfer, Daniel Graziotin, Ivan Bogicevic, Stefan Wagner, and Jasmin Ramadani. 2018. Poster: Communication in Open-Source Projects–End of the E-mail Era?. In *ICSE '18 Companion: 40th International Conference on Software Engineering Companion, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3183440.3194951

1 INTRODUCTION AND RELATED WORK

In open-source software (OSS) projects, which are characterized by distributed environments, a good communication system is of particular importance. Team members have to coordinate work, discuss tasks, solve problems, make decisions, and manage their projects continuously, over different time zones [1].

OSS stakeholders use several *types of communication channels*, i.e., electronic ways of communication such as mailing lists, forums, issue trackers, and chat systems as a practical way of exchanging information over large distances and coordinating work [3].

Studies show that mailing lists are the heart of any project communication [4], thus they should be the center of attention, but discussion on development aspects is shifting to the source code repository's issue system [3] or modern e-mail replacement systems such as Slack and Gitter [2].

There is work about which communication channels software developers use and which of them they think are important. Storey et al. [5] collected data from a survey of 1449 developers on what communication channels they use, which are the most important ones, and what challenges they face. Their results show which channels developers use for which development activity, e.g., to find answers. However, the results only show which channel is used for which activity but not the distribution of channels between repositories and other activities that might be done by other stakeholders.

We identified a gap in the current understanding of which tools are effectively employed by OSS projects overall. In this paper, we report an automated census that let us estimate the population of communication channels among GitHub software projects.

Research Question: *Which types of communication channels are used in open-source projects?*

2 METHOD

To provide a meaningful census of communication tools and exchanged information, we defined our population as those GitHub-hosted projects with the following characteristics: (1) active in the last six months (January 2017 to June 2017), (2) having at least 20 code commits, and (3) having at least 10 contributors.

We gathered the population by querying the *GitHub Archive*, thus obtaining $13,757,509$ active projects after removing duplicates. We randomly sampled the population to check against our inclusion criteria, and we set a sample size of $n = 400$ random projects for an error margin to 5% and confidence level to 95%.

Then, we developed a Python tool using a series of regular expressions for mining the communication channels of the *description artifacts* (projects' description, README and Wiki files). The tool uses the GitHub APIs for querying the description artifacts, and it

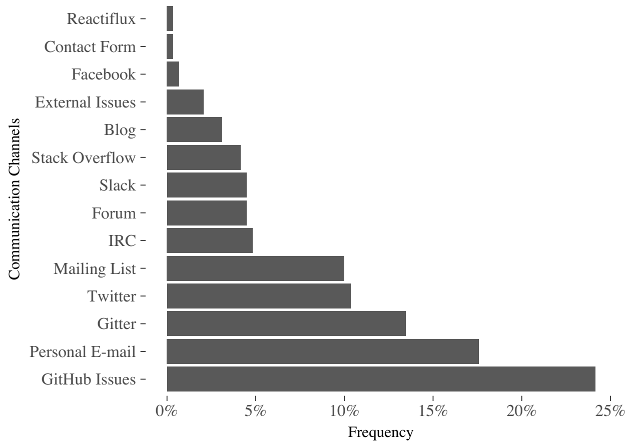

Figure 1: Types of communication channels of all communication channels in GitHub projects

runs a series of regular expressions to identify the communication channels. The expressions match how external channels are referenced in the description artifacts (e.g., IRC could be mentioned in ways including *IRC: #<channel>* or *irc.freenode.net: #<channel>*).

We followed a systematic process of randomly sampling $n = 40$ projects for developing and testing our regular expressions. We compared the tool classifications with those of a human rater and repeated the cycle until reaching a threshold of 86% of channels with respect to those found by a human, thus yielding a 14% false negative rate. Furthermore, the tool had a 18% false positive rate by flagging channels that were not actually channels.

3 RESULTS

The tool identified 187 projects (46.7% of the 400 projects) using 290 communication channels (min=1.0, 1st quartile=1.0, median=1.0, mean=1.15, standard deviation=0.96, 3rd quartile=2.0, max=6.0).

***Estimation:** Less than half of the projects on GitHub employ communication channels.*

Figure 1 shows the identified channels and their usage.

We found the following communication channels, in order of frequency: GitHub Issues (24.1% [1]), personal e-mail (17.6%), Gitter (13.4%), Twitter, mailing lists (10%), IRC, forums, Slack, Stack Overflow, blogs, external issue trackers, Facebook, contact forms, and Reactiflux.

Gitter usage was far higher than expected (used in 9.7% of the $n = 400$ GitHub projects), surpassing the use of mailing lists (used in 7.2% of the $n = 400$ GitHub projects).

Regarding the foreseen but unconfirmed decline of mailing lists [2–4], we observe that modern e-mail replacement chat systems such as Gitter, Slack, and Reactiflux are replacing the use of mailing lists, with a combined active usage of 18.2% of communication instances (and used in 13.25% of the $n = 400$ GitHub projects) versus the 10% usage of mailing lists (and used in 7.2% of the $n = 400$ GitHub projects).

[1]All percentages are relative to the $n = 290$ of communication channels

***Estimation:** Mailing lists are being replaced by modern enterprise chat systems in OSS development.*

4 DISCUSSION AND CONCLUSION

In this study, we performed an automatized analysis of a representative sample of open-source projects for describing the current communication channels in use.

We show that:

- Only half of the projects use externally visible communication channels.
- GitHub Issues, personal e-mail, Gitter, Twitter, and mailing lists are the most popular channels, in that order.
- Mailing lists appear to be losing market share in favor of modern, enterprise chat systems such as Gitter and Slack.

We suggest future work to seek support for our estimation that half of all open-source projects do not employ observable communication channels and to understand if their omission has an effect on the efficiency or success of the projects.

Finally, we see that mailing lists are likely to disappear over time in favor of modern chat replacements, which have a combined usage of 18.2% among communication channels in GitHub projects (estimated adoption in 13.25% of projects). Studies on the communication in open-source projects should not focus on mailing lists only but need to take the diversity of communication channels into account.

More specifically, we are adding to the growing evidence that mailing lists are diminishing in favor of enterprise chat systems such as Gitter, Slack, and Reactiflux.

Our results are in line with the recent suggestions [2] that enterprise chats are playing an increasingly significant role in software engineering. Future studies should attempt to go deep and classify the information exchanges in enterprise chat systems both in terms of types and frequency.

ACKNOWLEDGMENT

Daniel Graziotin has been supported by the Alexander von Humboldt (AvH) Foundation.

REFERENCES

[1] Christian Bird, David Pattison, Raissa D'Souza, Vladimir Filkov, and Premkumar Devanbu. 2008. Latent Social Structure in Open Source Projects. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT '08/FSE-16).* ACM, New York, NY, USA, 24–35. https://doi.org/10.1145/1453101.1453107

[2] Darren Gergle, Meredith Ringel Morris, Pernille BjÃÿrn, and Joseph Konstan (Eds.). 2016. *Why Developers Are Slacking Off: Understanding How Software Teams Use Slack.* Vol. the 19th ACM Conference. ACM Press, New York, New York, USA.

[3] Anja Guzzi, Alberto Bacchelli, Michele Lanza, Martin Pinzger, and Arie van Deursen. 2013. Communication in Open Source Software Development Mailing Lists. In *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13).* IEEE Press, Piscataway, NJ, USA, 277–286. http://dl.acm.org/citation.cfm?id=2487085.2487139

[4] Emad Shihab, Nicolas Bettenburg, Bram Adams, and Ahmed E. Hassan. 2010. On the Central Role of Mailing Lists in Open Source Projects: An Exploratory Study. Springer Berlin Heidelberg, Berlin, Heidelberg, 91–103.

[5] M. A. Storey, A. Zagalsky, F. F. Filho, L. Singer, and D. M. German. 2017. How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development. *IEEE Transactions on Software Engineering* 43, 2 (Feb 2017), 185–204. https://doi.org/10.1109/TSE.2016.2584053