Interpersonal Trust in OSS: Exploring Dimensions of Trust in GitHub Pull Requests

Amirali Sajadi
Drexel University
Philadelphia, Pennsylvania, USA
amirali.sajadi@drexel.edu

Kostadin Damevski
Virginia Commonwealth University
Richmond, Virginia, USA
kdamevski@vcu.edu

Preetha Chatterjee
Drexel University
Philadelphia, Pennsylvania, USA
preetha.chatterjee@drexel.edu

Abstract—Interpersonal trust plays a crucial role in facilitating collaborative tasks, such as software development. While previous research recognizes the significance of trust in an organizational setting, there is a lack of understanding in how trust is exhibited in OSS distributed teams, where there is an absence of direct, in-person communications. To foster trust and collaboration in OSS teams, we need to understand what trust is and how it is exhibited in written developer communications (e.g., pull requests, chats). In this paper, we first investigate various dimensions of trust to identify the ways trusting behavior can be observed in OSS. Next, we sample a set of 100 GitHub pull requests from Apache Software Foundation (ASF) projects, to analyze and demonstrate how each dimension of trust can be exhibited. Our findings provide preliminary insights into cues that might be helpful to automatically assess team dynamics and establish interpersonal trust in OSS teams, leading to successful and sustainable OSS.

Index Terms—trust, open source software, pull requests

I. INTRODUCTION

Although Open Source Software (OSS) has revolutionized the software development industry, stakeholders still face challenges in increasing the likelihood of new projects' success and maintaining activity in existing projects [12], [44]. Studies suggest the inexorable importance of the health of the project participants towards the sustainability of OSS projects [46]. For instance, McDonald and Goggins found that project success in OSS is attributed more to contributor growth and community involvement than to source code metrics [25]. Issues stemming from person/team dynamics, e.g., conflict between developers, is identified as one of the key reasons for failures in OSS projects [8]. Interpersonal trust could facilitate collaboration, and lower the conflicts among team members [4], [5]. Additionally, given the evidence suggesting that the newcomers' decision to leave OSS projects is influenced by their interactions with other team members [39], there is an opportunity in studying how building trust in early stages can help with the retention of newcomers in OSS.

Having been approached from a broad range of disciplines, human trust has been studied in various contexts [35]; but, the role of interpersonal trust in the context of software engineering has yet to be sufficiently examined and explored. Trust has been shown to influence the ability of a team to have better cohesion and more cooperation [1], to affect the developers in decision making [30] and to retain new contributors in projects [20]. Furthermore, trust among team

TABLE I
TRUST, AS DEFINED IN THE CONTEXT OF TEAM MANAGEMENT.

"the overall willingness of virtual team members to rely on one another that results from the aggregate of potential trust dimensions that are achieved through socio-emotional and task process and supported by technology capabilities." Mitchell & Zigurs [27]
"the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other party will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party" Mayer et al. [24]
"a positive psychological state (cognitive and emotional) of a trustor (person who can trust/distrust) towards a trustee (person who can be trusted/distrusted) comprising of trustor's positive expectations of the intentions and future behaviour of the trustee, leading to a willingness to display trusting behaviour in a specific context." Rusman et al. [35]
"an actor's expectation of the other actors' capability, goodwill and self-reference visible in mutually beneficial behaviour enabling cooperation under risk" Henttonen & Blomqvist [13]

members has been used to determine the overall efficiency and the success of OSS development teams, while reduced levels of trust have been associated with a lack of unity and less collaboration [5].

Researchers have investigated trust, and how different types of trust (e.g., cognitive trust, dependency trust) are exhibited in an organizational setting [4], [18], [35], [37], [42]. Trust in the context of team management has been defined by multiple studies, some of which are listed in Table I. However, there is no consensus on a universal definition of trust [27], [34]. To foster trust among developers and better manage teams, we first require a deeper understanding of what trust is and how it is manifested in the software development environments.

In this paper, we address that gap by exploring how interpersonal trust is exhibited in OSS. First, we leverage theories from psychology and organizational behavior to investigate dimensions of trust. Each of the trust dimensions we examined is exhibited in specific types of interpersonal interactions, while the overall trust between a pair of individuals is the accumulation of all these dimensions [27]. With the goal of representing instances of trust in OSS, we investigate pull requests on GitHub and create a mapping between the trust dimensions and activities in pull requests (Section II). Next, we provide qualitative and quantitative support for the dimensions of trust we observed in a set of 100 GitHub pull requests (Section IV). Our results provide preliminary indications regarding the dimensions of trust that are exhibited in OSS, specifically in pull requests, and also provide insights into what

cues might help in automatically extracting that information. The overarching goal of this study is to bring to attention the importance of interpersonal trust in OSS sustainability.

II. Trust dimensions

We select from the trust dimensions aggregated in the survey by Mitchell and Zigurs [27] with the goal of representing instances of trust in OSS. Using the definition of each trust dimension, we selected the dimensions of trust that are not conflicting or overlapping with one another, and are applicable to OSS projects on GitHub. The trust dimensions we selected are as follows:

Action-based trust: One of the characteristics of high-trust teams is frequent and detailed communication [17]. While feedback in teams with lower levels of trust is usually limited to a few words (e.g., 'ok', 'looks good'), teams with high-trust provide detailed feedback to each other's contributions.

Commitment trust: Commitment trust is tied to the extent to which members of a team perceive themselves as a part of a team [28]. The more engaged and trusted the developers are, the more committed they are to the project.

Competence trust: The competence and the overall abilities of an individual to perform certain tasks within a specific context is closely related to how trustworthy they may seem to others [19], [35]. The trust inspired by one's perceived competence in a virtual team often depends on the data that can be publicly observed for each user.

Institutional trust: People who work in the same environment led by the same set of rules are more likely to trust one another [38]. Belonging to certain known group or organization may lend a developer a degree of institutional trust.

Personality-based trust: An individual's general tendency to trust others, which is independent of external factors (e.g., the context), defines their personality-based trust. Those with a higher propensity to trust are more likely to trust others' ideas or contributions.

Transferred Trust: Most people view individuals recommended by someone they trust in a positive light. This positive attitude toward the recommended person is called transferred trust. With transferred trust, the individual extends her trust from the person she already knows to the recommended party.

III. Methodology

In order to understand trust in OSS projects and to perform our analysis, we focus on GitHub pull requests as they are a natural place to look for trust, specifically for the dimensions of trust we selected (in Section II). Having considered the data available on GitHub pull requests, we examined the features that support pull requests and the type of developer interactions they tend to elicit. Doing so allowed us to look for clear signals of trust between the participants (i.e., contributors and reviewers). More specifically, we considered the following types of information to understand how the trust dimensions are exhibited:

- *Pull request comments:* pull request comments are likely to contain valuable insight into the relationships of developers interacting with one another.
- *Pull request metadata:* For each pull request, we examined the following information: pull request creator, closer, the state of the pull request (accepted or rejected), the state of the reviews (approved, commented, changes requested), the labels, and the type of contribution (code or documentation).
- *User profile data:* A GitHub user's publicly available profile data can influence their social interactions and the trust that develops between them and other project participants. We examined the following data: 1) the number of followers, 2) the names of GitHub organizations the user belongs to, 3) previous actions in the repository of interest.

To identify trust in the pull requests in our dataset, we adopted a thematic approach and followed the steps recommended by Nowell et al. [29]. The process of analyzing the data was done iteratively and reflectively, alternating between different phases of analysis such as memoing, identifying themes, and refining, as recommended in qualitative analysis [22]. First, in order to better understand the interactions of the involved developers and the circumstances under which they have communicated, all the pull request comments and reviews were carefully read while keeping in mind the actions taken by the users to address the pull requests. Next, we used all the available data, including text, the background of the users (as available on their GitHub profiles), and the collected statistical data, to generate our initial set of mappings of trust dimensions with GitHub pull requests. These mappings indicated the presence or lack of a trust dimension in a data instance. We also made notes on the possible reasons for the pull request rejections as well as any relevant fact that could become useful in the following steps. We repeated the process multiple times, adding to our notes and observations, and reevaluating the mappings in each iteration.

In, Table II we show the result of this process, i.e., our mapping between the trust dimensions and the available data surrounding GitHub's pull requests.

IV. Preliminary Study

In this section, we discuss our observations for the dimensions of trust we examined in a sample of GitHub pull requests.

Data Selection. We selected four active repositories (`superset`, `beam`, `airflow`, and `dubbo`) from Yin et al.'s dataset of 269 Apache Software Foundation (ASF) projects [45]. Each repository in our dataset contains at least 5900 stars, 400 contributors, and 5000 pull requests. For the purpose of this study, we randomly sampled a set of 100 pull requests, 25 from each repository. All four repositories we examined contain predominantly accepted pull requests and few rejected pull requests. Therefore, to investigate the differences in the levels of trust exhibited in accepted vs. rejected pull requests, we sampled 75% accepted and 25% rejected pull requests.

Observations. We examined the pull requests in our sample for instances of trust (expressed through the dimensions in

TABLE II
TRUST DIMENSIONS IN GITHUB PULL REQUESTS.

This dimension of trust develops from...	For Pull Requests, this dimension of trust can be observed in...
Action-based trust – "a process based on or promoted by fast and frequent feedback with minimal delay." [3], [27].	The frequency and quality of comments and reviews of contributed pull requests can form a basis for action-based trust between the contributor and the reviewer.
Commitment trust – "a process based on contractual agreements (formal or psychological) between parties who have an expectation of mutual benefit derived from cooperative relations." [27]	Reviewing pull requests when requested or addressing a reviewer's comment indicates the existence of commitment trust between the contributor and reviewer.
Competence trust – "a process based on perceptions of another's competence to carry out the tasks that need to be performed and is based on an attitude of respect for the abilities of the trustee to complete their share of the job at hand." [27]	Developers' past actions in the repository, their overall activity, and their status on GitHub (e.g., previous pull requests, role in the project, level of activity, number of followers, etc.) influence the trustor's perception of the trustee's competence.
Institutional trust – "a process guided by the norms and rules of institutions (such as organizations) and based on formal institutional arrangements such as contracts, sanctions, or legal procedures." [27]	Developers associated with the same organizations (or working for the same company) are more likely to trust each other on the basis of their affiliations with these organizations.
Personality-based trust – "a developmental process that occurs during infancy when a person seeks help from caretakers and that results in a general propensity to trust others." [27]	Project developers/maintainers who tend to accept a high number of pull requests from various users have a higher propensity to trust people and their work. On the other hand, those who tend to reject more pull requests may have a lower propensity to trust others.
Transferred trust – "a process that may occur when the trustor knows and trusts a person or the institution that recommends the trustee." [3], [27]	In some cases, well-established members of the project can recommend the work of new and less well-known contributors. This leads to other project participants who trust them to now trust the new contributor.

Table II). In Figure 1, we visualize two example pull requests where: (a) one is exhibiting high levels of trust, (b) one is exhibiting low levels of trust. We report our observations on how each trust dimension manifests in pull requests below:

1) Action-based trust: We found numerous instances of developer interactions in pull requests that resemble action-based trust. We observe that, compared to rejected pull requests, accepted pull requests generate more social interactions (e.g., comments) in a shorter period of time. Additionally, the creators of accepted pull requests are often more responsive to other users' questions. They actively try to explain or improve the quality of the contribution by asking for feedback and reviews from other developers. It is therefore more common to see multiple new commits and revisions in accepted pull requests, compared to the rejected ones. In our dataset, the average comment frequency (the number of comments in a day) is 4 for the accepted pull requests, and ∼1.25 for rejected pull requests. 75 out of 100 pull requests involved additional commits that were made as a result of developers' interactions. 66 of the accepted pull requests contributed to this number, while only 9 rejected pull requests contained additional commits made as a result of further communication.

Overall, the number and frequency of the comments and the conversations leading to new commits have been used as indicators of action-based trust in the pull requests. For instance, the pull request in Figure 1(a) has generated a high frequency of comments and created discussions that have resulted in revisions to the code. The pull request in Figure 1(b), however, has not received as much attention in the comments section, nor has it received any reviews that would aid the contribution to be further made suitable for merging.

2) Commitment trust: Given the regularity and the emphasis on the reviews in pull requests, we observe several instances of users requesting and making pull request reviews in our dataset. It is uncommon for a requested user to make reviews after another requested user has already reviewed the work. This behavior could be attributed to the high volume of requests and the fact that the pull requests may not need

multiple review approvals to get merged. When developers ask for reviews from multiple users, most often at least one of the requested users provides a review. 35 out of 100 pull requests in our dataset (32 accepted and 3 rejected pull requests) had at least one response to the requested reviews.

The pull request creator in Figure 1(a) requests four developers for reviews and receives a detailed review from one. On the other hand, in Figure 1(b) the pull requester does not request a review or respond to the other user.

3) Competence trust: Among the different profile attributes that we considered (number of followers, number of pull requests made and their acceptance/rejection rate in the last 1000 pull requests, write/read permission in the repository), the perceived trustworthiness of a user is found to be closely related to their previous actions. Out of the 24 rejected pull requests, 9 were made by users who had not previously made any accepted pull requests in the project. Additionally, 4 of these 9 users were making their first pull requests in the project, while the other 5 had only made 1 rejected pull request prior to that point.

In Figure 1(a), `ashb`, the pull requester, is a member of the project with write access in the repository, has a noticeable number of followers, and a high rate of acceptance for his pull requests; all qualities that would indicate competence trust. The pull requester in Figure 1(b) has also made a considerable number of accepted pull request in the project, but compared to the requester in Figure 1(a), he has a substantially less number of followers and is not a project member.

4) Institutional trust: Inspecting the overlap between the developers' GitHub organizations, we observed that pull requests involving users with same affiliations has often led to successful contributions to the project (e.g., contributions that were not rejected on the spot). It should be noted that although GitHub organizations can help us detect a user's associations, not all organizational affiliations are available on GitHub. Additionally, an user can trust others based on their affiliations with certain organizations without being part of that organization. Out of the 25 rejected pull requests in
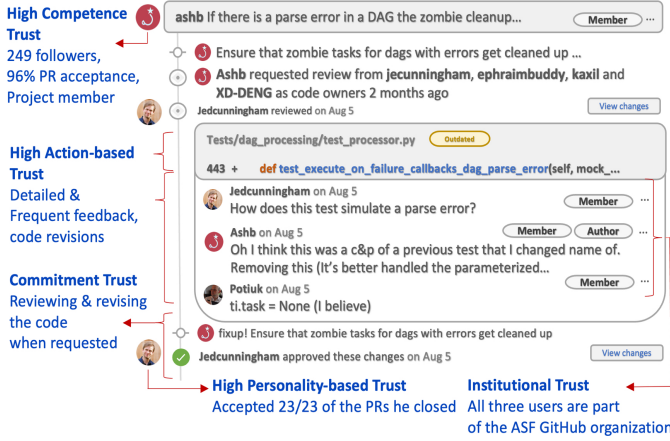
(a) A pull request interaction with high levels of trust.
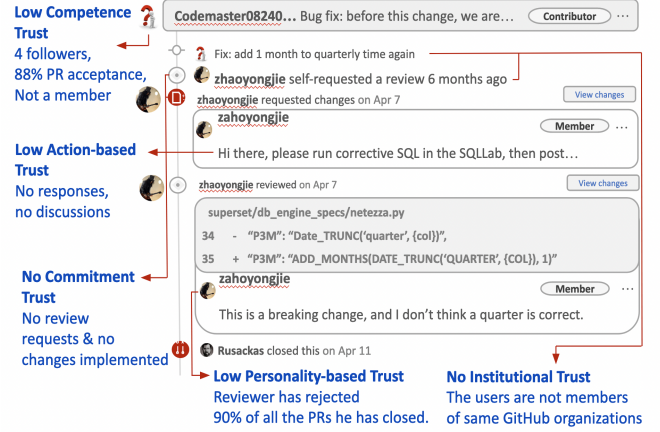
(b) A pull request interaction with low levels of trust.

Fig. 1. Examples of Trust exhibited in GitHub pull requests.

our dataset, we did not observe any commenters or reviewers sharing the same organizations as the pull requesters. For accepted pull requests, we observed institutional trust in up to 11 pull requests out of the 19 accepted pull requests in each repository using the GitHub organizations.

The pull requester and both of the reviewers in Figure 1(a) are parts of the "The Apache Software Foundation" GitHub organization. No such cues for institutional trust is observed in the pull request in Figure 1(b).

5) Personality-based trust: We can estimate the personality-based trust of a developer using the number of pull requests rejected or accepted. Surprisingly, most developers who closed pull requests have almost always accepted them. Specifically, 79 out of 100 pull requests had a closer or at least one reviewer who had accepted all the pull requests she had closed. For instance, we found that a specific user closed 272 pull requests, all of which were accepted. This might be due to the fact that each of the projects we examined are managed by small groups of active developers who contribute frequently and have most of their pull requests accepted. This pull request acceptance behavior and team dynamics is common across many OSS repositories [39]. More interestingly, we found that there are a set of users who predominantly reject the pull requests they close. Even though these users interact with other developers in different contexts (e.g., merged pull requests), the pull requests they close are almost always rejected.

The reviewers `jedcunningham` and `potuik` in Figure 1(a), almost always accept the pull requests they close, which we interpret as a sign of a high propensity to trust other developers' work. Conversely, `zhaoyongjie`, the reviewer in Figure 1(b), has only accepted approximately 10% of all the pull requests they have closed (10 out of 99).

6) Transferred trust: We observed only one instance of transferred trust in our dataset. A significant contribution by a newcomer caused one of the project members to seek another team member's opinion on the work[1]. The team member responded by introducing the pull requester as a new member of their team and assured the other party of the work's quality; *"Syed is a new member of our team, we already reviewed his work :) Thanks for reaching out though"*. Overall, we require more data to better understand and observe this dimension of trust and the way it is exhibited on GitHub.

V. RELATED WORK

From human trust in automation systems to designing better trust management [43] and e-commerce systems that find the best matches between customers and sellers based on trust [14], prior research has approached the concept of trust from various angles and in different capacities. Trust in software, systems and the ties between trust, privacy, and security are some of the ways trust has been studied in a software-related context [21], [31]. Some studies have examined trust as an interpersonal concept, trying to shed light on how trust is developed and how it affects the developers in software engineering and open source teams [2], [9], [17], [32], [33], [36]. Some of these studies use questionnaires or interviews to identify and understand trust in an organization and teams, while others have proposed methods to estimate the levels of trust in indirect ways. Wang & Redmiles, for instance, collect data about trust through interviews to gain insight into the development of trust among software developers [41], while da Cruz et al. [11] puts forth a framework, meant to automatically detect trust among OSS project members. Venigalla & Chimalakonda use an emotion analysis tool, and Sapkota et al. create a framework to automatically identify trust in Github [37], [40]. Another study tries to evaluate developers' propensity to trust others by making use of IBM Watson's Tone Analyzer and the *Big-Five personality* model [10], a famous taxonomy of human personality traits, to find correlations between one's tendency to trust others and the chances of her accepting or rejecting pull requests. Likewise, Iyer et al.

---
[1]https://github.com/apache/airflow/pull/25324

draws conclusions about propensity to trust in developers and the ways in which it can affect the chances of a pull request to get accepted or rejected [16]. Lastly, Calefto & Lanubile use sentiment analysis to build upon Wang & Redmiles's results and estimate trust among developers [4]. Apart from trust, researchers have analyzed and designed techniques for the detection of various emotions such as anger, stress, toxicity in communications related to software development [6], [7], [15], [23], [26]. Compared to other work, we have investigated trust in OSS at a different granularity, by leveraging dimensions of trust based on theories in organizational behavior and psychology.

VI. THREATS TO VALIDITY

Construct Validity. The complex nature of interpersonal trust does not allow us to precisely determine the level of trust or lack thereof among individuals based on a limited number of written interactions. This limitation poses an unavoidable threat to the construct validity of the our study. Building upon prior theoretical work, we have attempted to address the inherent complications of identifying interpersonal trust in GitHub's OSS projects. We use the same definitions of trust dimensions that have been proposed in the existing body of literature to ensure that the studied measures are, in fact, indicative of the answers we seek.

Internal Validity. The human interactions among those involved with any OSS project may not be limited to pull requests. Even within the GitHub platform itself, there are other ways in which developers typically interact with one another. Furthermore, the developers can contact each another through different means and platforms, e.g., email or Slack. It is even possible that some developers know each other personally and have interactions outside the development environment. These factors can all affect the conclusions from our preliminary study. To mitigate this threat, we have limited our subjects of study to ASF projects. This measure limited the communication channels outside GitHub that the developers can use to contact each other as ASF projects follow specific communication guidelines. The ASF project members use mailing lists and GitHub features such as Discussions and Issues to communicate. Examining all these other communications is beyond the scope of this study and remains to be further explored in future research.

External Validity. In presenting the ways in which trust is manifested on GitHub, we have used a dataset of 100 randomly selected pull requests from four active repositories, which is a threat to generalizability. To mitigate this threat, we selected the subjects of our study from 4 popular OSS projects. Scaling to larger and more diverse datasets might lead to different study observations. We hope to assess the generalizability of our results by analyzing a larger dataset as a part of our future work.

VII. CONCLUSION AND FUTURE WORK

The success of an OSS team effort is heavily influenced by the interpersonal dynamics of the developers. In this work,

we sought to elucidate the crucial role of interpersonal trust among the OSS developers and to identify and illustrate this trust within the development context through our mapping of trust dimension to the GitHub interaction (Table II). The end goal of our research is to design an automatic system to detect and analyze trust in OSS teams. Such automation would allow the monitoring of a team's level of trust and timely interventions for project managers. Furthermore, the measurements of trust can help us uncover ties between trust and the overall performance of a team and thus aid the tracking the progress of the project throughout the software development lifecycle.

Some of the cues we have used to determine trust can be quantitatively measured, leading to automatically computable measures (e.g., measuring *Competence trust* from developer profile). However, other signals of trust, such as the textual data analyzed to detect *Transferred trust* are not easily computable and might require more sophisticated natural language analyses. Of course, a larger study is needed to gain a deeper understanding and overcome these challenges. As a part of a future study, we also plan to expand the dataset to include the ASF mailing list, which offers a rich source of data for developer communications beyond pull requests. The email addresses of the members can also enable us to detect members of the same team and better identify *Institutional trust*. Our replication package is publicly available here.

VIII. ACKNOWLEDGEMENTS

REFERENCES

[1] Cristina Acedo-Carmona and Antoni Gomila. Personal trust increases cooperation beyond general trust. *PLoS One*, 9(8):e105559, 2014.

[2] Ban Al-Ani, Hiroko Wilensky, David Redmiles, and Erik Simmons. An understanding of the role of trust in knowledge seeking and acceptance practices in distributed development teams. In *2011 IEEE Sixth International Conference on Global Software Engineering*, pages 25–34. IEEE, 2011.

[3] Patricia M Alexander. Teamwork, time, trust and information. In *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 65–74, 2002.

[4] Fabio Calefato and Filippo Lanubile. Affective trust as a predictor of successful collaboration in distributed software projects. In *2016 IEEE/ACM 1st International Workshop on Emotional Awareness in Software Engineering (SEmotion)*, pages 3–5. IEEE, 2016.

[5] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. A preliminary analysis on the effects of propensity to trust in distributed software development. In *2017 IEEE 12th international conference on global software engineering (ICGSE)*, pages 56–60. IEEE, 2017.

[6] P. Chatterjee, K. Damevski, L. Pollock, V. Augustine, and N.A. Kraft. Exploratory Study of Slack Q&A Chats as a Mining Source for Software Engineering Tools. In *Proceedings of the 16th International Conference on Mining Software Repositories (MSR'19)*, May 2019.

[7] Preetha Chatterjee, Kostadin Damevski, and Lori Pollock. Automatic extraction of opinion-based Q&A from online developer chats. In *Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1260–1272, 2021.

[8] Jailton Coelho and Marco Tulio Valente. Why modern open source projects fail. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2017, page 186–196, New York, NY, USA, 2017. Association for Computing Machinery.

[9] Gail Corbitt, Lorraine R Gardiner, and Lauren K Wright. A comparison of team developmental stages, trust and performance for virtual versus face-to-face teams. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, pages 8–pp. IEEE, 2004.

[10] Paul T Costa Jr and Robert R McCrae. The five-factor model, five-factor theory, and interpersonal psychology. *Handbook of interpersonal psychology: Theory, research, assessment, and therapeutic interventions*, pages 91–104, 2010.

[11] Guilherme Augusto Maldonado da Cruz, Elisa Hatsue Moriya-Huzita, and Valéria Delisandra Feltrim. Arsenal-gsd: A framework for trust estimation in virtual teams based on sentiment analysis. *Information and software technology*, 95:46–61, 2018.

[12] Marco Gerosa, Igor Wiese, Bianca Trinkenreich, Georg Link, Gregorio Robles, Christoph Treude, Igor Steinmacher, and Anita Sarma. The shifting sands of motivation: Revisiting what drives contributors in open source. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1046–1058, 2021.

[13] Kaisa Henttonen and Kirsimarja Blomqvist. Managing distance in a global virtual team: the evolution of trust through technology-mediated relational communication. *Strategic Change*, 14(2):107–119, 2005.

[14] Kevin Anthony Hoff and Masooda Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors*, 57(3):407–434, 2015.

[15] Mia Mohammad Imran, Yashasvi Jain, Preetha Chatterjee, and Kostadin Damevski. Data augmentation for improving emotion recognition in software engineering communication. In *37th IEEE/ACM International Conference on Automated Software Engineering*, ASE22, New York, NY, USA, 2023. Association for Computing Machinery.

[16] Rahul N Iyer, S Alex Yun, Meiyappan Nagappan, and Jesse Hoey. Effects of personality traits on pull request acceptance. *IEEE Transactions on Software Engineering*, 47(11):2632–2643, 2019.

[17] Sirkka L Jarvenpaa, Kathleen Knoll, and Dorothy E Leidner. Is anybody out there? antecedents of trust in global virtual teams. *Journal of management information systems*, 14(4):29–64, 1998.

[18] Mohammad Saud Khan. Role of trust and relationships in geographically distributed teams: exploratory study on development sector. *International Journal of Networking and Virtual Organisations*, 10(1):40–58, 2012.

[19] Feng-yang Kuo and Chia-ping Yu. An exploratory study of trust dynamics in work-oriented virtual teams. *Journal of Computer-Mediated Communication*, 14(4):823–854, 2009.

[20] Michael S Lane, Glen van der Vyver, Prajwal Basnet, and Srecko Howard. Interpretative insights into interpersonal trust and effectiveness of virtual communities of open source software (oss) developers. In *Proceedings of the 15th Australasian Conference on Information Systems (ACIS 2004)*. University of Tasmania, 2004.

[21] Luigi Lavazza, Sandro Morasca, Davide Taibi, and Davide Tosi. Predicting oss trustworthiness on the basis of elementary code assessment. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–4, 2010.

[22] Sarah Lewis. Qualitative inquiry and research design: Choosing among five approaches. *Health Promotion Practice*, 16(4):473–475, 2015.

[23] Mika Mäntylä, Bram Adams, Giuseppe Destefanis, Daniel Graziotin, and Marco Ortu. Mining valence, arousal, and dominance: Possibilities for detecting burnout and productivity? In *Proceedings of the 13th International Conference on Mining Software Repositories*, MSR '16, page 247–258, New York, NY, USA, 2016. Association for Computing Machinery.

[24] Roger C Mayer, James H Davis, and F David Schoorman. An integrative model of organizational trust. *Academy of management review*, 20(3):709–734, 1995.

[25] Nora McDonald and Sean Goggins. Performance and participation in open source software on github. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, page 139–144, 2013.

[26] Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilescu, and Christian Kästner. "did you miss my comment or what?" understanding toxicity in open source discussions. In *In 44th International Conference on Software Engineering (ICSE'22)*, 2022.

[27] Alanah Mitchell and Ilze Zigurs. Trust in virtual teams: solved or still a mystery? *Acm Sigmis Database: The Database for Advances in Information Systems*, 40(3):61–83, 2009.

[28] Sue Newell, Gary David, and Donald Chand. Exploring trust among globally distributed work teams. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 246c–246c. IEEE, 2007.

[29] Lorelli S Nowell, Jill M Norris, Deborah E White, and Nancy J Moules. Thematic analysis: Striving to meet the trustworthiness criteria. *International journal of qualitative methods*, 16(1):1609406917733847, 2017.

[30] Heikki Orsila, Jaco Geldenhuys, Anna Ruokonen, and Imed Hammouda. Trust issues in open source software development. In *Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010*, pages 9–12, 2009.

[31] Ipek Ozkaya. Understanding and building trust in software systems. *IEEE Software*, 39(3):3–6, 2022.

[32] Souren Paul and Fang He. Time pressure, cultural diversity, psychological factors, and information sharing in short duration virtual teams. In *2012 45th Hawaii International Conference on System Sciences*, pages 149–158. IEEE, 2012.

[33] Ramón Rico, Carlos-Maria Alcover, Miriam Sánchez-Manzanares, and Francisco Gil. The joint relationships of communication behaviors and task interdependence on trust building and change in virtual project teams. *Social science information*, 48(2):229–255, 2009.

[34] Denise M Rousseau, Sim B Sitkin, Ronald S Burt, and Colin Camerer. Not so different after all: A cross-discipline view of trust. *Academy of management review*, 23(3):393–404, 1998.

[35] Ellen Rusman, Jan Van Bruggen, Peter Sloep, and Rob Koper. Fostering trust in virtual project teams: Towards a design framework grounded in a trustworthiness antecedents (twan) schema. *International Journal of Human-Computer Studies*, 68(11):834–850, 2010.

[36] Ellen Rusman, Jan Van Bruggen, Peter Sloep, Martin Valcke, and Rob Koper. Can i trust you?: Profile elements that inform first impressions of trustworthiness in virtual project teams. *International Journal of Information Technology Project Management (IJITPM)*, 3(1):15–35, 2012.

[37] Hitesh Sapkota, Pradeep K Murukannaiah, and Yi Wang. A network-centric approach for estimating trust between open source software developers. *Plos one*, 14(12):e0226281, 2019.

[38] Saonee Sarker, Joseph S Valacich, and Suprateek Sarker. Virtual team trust: Instrument development and validation in an is educational environment. *Information Resources Management Journal (IRMJ)*, 16(2):35–55, 2003.

[39] Igor Steinmacher, Igor Wiese, Ana Paula Chaves, and Marco Aurélio Gerosa. Why do newcomers abandon open source software projects? In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 25–32. IEEE, 2013.

[40] Akhila Sri Manasa Venigalla and Sridhar Chimalakonda. Understanding emotions of developer community towards software documentation. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 87–91. IEEE, 2021.

[41] Yi Wang and David Redmiles. Cheap talk, cooperation, and trust in global software engineering. *Empirical Software Engineering*, 21(6):2233–2267, 2016.

[42] Adam Wierzbicki. *Trust and fairness in open, distributed systems*, volume 298. Springer, 2010.

[43] Adam Wierzbicki, Tomasz Kaszuba, Radoslaw Nielek, Paulina Adamska, and Anwitaman Datta. Improving computational trust representation based on internet auction traces. *Decision Support Systems*, 54(2):929–940, 2013.

[44] Likang Yin, Zhunagzhi Chen, Qi Xuan, and Vladimir Filkov. Sustainability forecasting for apache incubator projects, 2021.

[45] Likang Yin, Zhiyuan Zhang, Qi Xuan, and Vladimir Filkov. Apache software foundation incubator project sustainability dataset. in 2021 ieee/acm 18th international conference on mining software repositories (msr)(2021).

[46] Marcelo Serrano Zanetti, Ingo Scholtes, Claudio Juan Tessone, and Frank Schweitzer. The rise and fall of a central contributor: Dynamics of social organization and performance in the gentoo community. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 49–56, 2013.