Good Fences Make Good Neighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells

Stefano Lambiase
slambiase@unisa.it
SeSa Lab – University of Salerno
Salerno, Italy

Gemma Catolino
g.catolino@tilburguniversity.edu
Jheronimus Academy of Data Science
's-Hertogenbosch, The Netherlands

Damian A. Tamburri
d.a.tamburri@tue.nl
Jheronimus Academy of Data Science
's-Hertogenbosch, The Netherlands

Alexander Serebrenik
a.serebrenik@tue.nl
Eindhoven University of Technology
Eindhoven, The Netherlands

Fabio Palomba
fpalomba@unisa.it
SeSa Lab – University of Salerno
Salerno, Italy

Filomena Ferrucci
fferrucci@unisa.it
SeSa Lab – University of Salerno
Salerno, Italy

ABSTRACT

Software development is *de facto* a social activity that often involves people from all places to join forces globally. In such common instances, project managers must face social challenges, e.g., personality conflicts and language barriers, which often amount literally to "culture shock". In this paper, we seek to analyze and illustrate how *cultural and geographical dispersion*—that is, how much a community is diverse in terms of its members' cultural attitudes and geographical collocation—influence the emergence of collaboration and communication problems in open-source communities, *a.k.a. community smells*, the socio-technical precursors of unforeseen, often nasty organizational conditions amounting collectively to the phenomenon called social debt. We perform an extensive empirical study on cultural characteristics of GitHub developers, and build a regression model relating the two types of dispersion—cultural and geographical—with the emergence of four types of community smells, i.e., *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*. Results indicate that cultural and geographical factors influence collaboration and communication within open-source communities, to an extent which incites—or even more interestingly mitigates, in some cases—*community smells*, e.g., *Lone Wolf*, in development teams. Managers can use these findings to address their own organizational structure and tentatively diagnose any nasty phenomena related to the conditions under study.

CCS CONCEPTS

• **Software and its engineering → Software organization and properties**; • **Social and professional topics → Cultural characteristics**; **Geographic characteristics**.

KEYWORDS

Global Software Engineering; Cultural Dispersion; Community Smells; Software Organizational Structures; Empirical Studies.

ICSE-SEIS'22, May 21–29, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9227-3/22/05…$15.00
https://doi.org/10.1145/3510458.3513015

ACM Reference Format:
Stefano Lambiase, Gemma Catolino, Damian A. Tamburri, Alexander Serebrenik, Fabio Palomba, and Filomena Ferrucci. 2022. Good Fences Make Good Neighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells. In *Software Engineering in Society (ICSE-SEIS'22), May 21–29, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3510458.3513015

LAY ABSTRACT

To what extent does the global and multi-cultural nature of software engineering influence software processes welfare? More specifically, does an increase in "globalization" of software activities negatively or positively influence known nasty effects common in the process of software construction? Rotating around these questions, this research finds that there is in fact evidence of the aforementioned influence but it does not provide for positive effects only. Specifically, a decrease of globalization does not necessarily bode positively on conditions such as lone developers working in an individualistic fashion—a phenomenon known as "lone wolf" effect—and other nasty organizational phenomena potentially slowing down or halting software construction and maintenance activities.

1 INTRODUCTION

Software engineering is, by nature, a social activity [14, 38, 51] that often involves organizations, managers, developers, and stakeholders from various places around the world [24, 33, 53] to share in collective action. This leads engineers and managers to face various challenges especially in terms of collaboration and communication, such as personality conflicts, language, and culture barriers.

Focusing on such social problems, previous work as focused on *social debt*—i.e., unforeseen project costs connected to a suboptimal development community [62, 63]—and in particular, phenomena known as *community smells* [61]—i.e., a set of socio-technical characteristics (e.g., high formality) and patterns (e.g., recurrent condescending behavior or rage-quitting)—which may lead to the emergence of social debt [47, 60, 61, 64]. Recently, the research community started exploring the diffusion and the impact of community smells, other than the factors that correlated their emergence [4, 5, 48]. For instance, several studies investigated how presence of women and gender diversity in development teams relate to community smells [21, 22, 55].

At the same time, Global Software Engineering (GSE)—whose core is the separation of development team in sub-communities, like separated by *"fences"*—has also studied the influence of cultural aspects on distributed software development [24, 58]. Researchers in such a field have concluded that Culture—the multi-faceted concept described by Hofstede as "the common characteristic that distinguishes the members of one human group from another" [37]—may be applied in terms of nations, regions, ethnic groups or subcommunities within organizations, with an influence going well beyond what can be observable and measurable. Although global software development teams benefit from cultural differences [33, 50], such differences can also lead to problems, specifically in collaboration and communication activities [18, 24, 28].

While previous research in Software Engineering has shown that cultural and physical distances may cause lower productivity, other than slowing down communication activities [24, 27, 28, 56], very little is known on how they impact sub-optimal communication and collaboration patterns like community smells. An improved understanding of these aspects may help researchers in quantifying the effects of cultural and geographical distances on socio-technical problems other than providing practitioners with mechanisms to monitor community health and take appropriate mitigation strategies. To this purpose, we use a subset of known community smells—i.e., *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*—already used in past literature to represent communication and collaboration problems and investigate the extent to which culture plays a role in their mediation [21]. Moreover, research on community smells is a young field of study, and there is a high need to identify possible factors able to influence the emergence of such patterns [64]. Seeing that culture and geographical distribution have been related to social problems, it is possible that they can influence the emergence of community smells.

Starting from these motivations, in this paper, we bridge this gap of knowledge by investigating how culture is related to communication and collaboration concerns—namely, community smells—of globally distributed software projects. To represent the cultural attitudes, we use the well-known ***Hofstede's 6-D framework*** [35, 37] that consists of six numerical dimensions (from zero to one hundred) able to model the cultural aspects of a country and its inhabitants, e.g., *Power Distance Index*. The motivations around the choice of Hofstede's framework are reported in Section 2.1. Specifically, we use the concept of *cultural dispersion*, i.e., how much a community presents members with different cultural attitudes. In order to operationalize the *geographical dispersion* of a development community, we used the spherical distance between the members of such a community. Precisely, we used the standard deviation of the set of spherical distances between each community member pair to obtain a reliable measurement. Furthermore, to operationalise communication and cooperation problems we exploit four known and detectable **community smells** [47, 60, 61, 64], i.e., *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*. Afterwards, we conduct a two-step empirical study featuring:

(1) A statistical analysis of 23 493 open-source communities hosted on GitHub; the analysis was aimed at studying how much such communities are culturally dispersed;
(2) The definition of four statistical models, one for each community smell, to assess the relationship between cultural and geographical dispersion—plus a selected set of control variables—and community smells.

The results of our study reveal that an increased cultural and geographical dispersion does in fact impact the emergence of all the considered community smells; interestingly, not all smells are affected in the same way. For instance, the presence of both individuals from individualistic and collectivist culture in the same team could lead to the emergence of a *Lone Wolf* effect.

These results indicate that project managers should consider cultural and geographical as first-class citizens not only when shaping their own offshoring activities but the same aspects need to become prime during the planning, initiation, and retrospective phases of the project; particularly, further research is needed in the way culture and its characteristics' affect differently one team member to others and in which—possibly "smelly"—circumstances. What is more, the connection with the mitigation or even counterinsurgency of specific community smells needs to be supported with specific analytics. Similarly, we conclude that smell-specific analytics would also be needed during the process of Application Lifecycle Management (ALM) of globally-distributed software workforces, to detect effective patterns of work as well as equally effective mitigations for known community smells.

2 BACKGROUND AND RELATED WORK

This section describes the background and related work that is the foundation for our contributions.

2.1 Culture in Software Engineering

Software development is, even more, a geographically distributed activity [33, 45]. As a result, *Global Software Engineering* (GSE) was born to address the challenges—mainly social—derived from such a nature [53, 58]. Of all the various problems, cultural ones arise as able to lead to the emergence of catastrophic situations [27, 56].

Kreitner et al. define culture as being: *"socially derived, taken for granted assumptions about how to act and think"* [40]. Undoubtedly, culture represent a complex topic, hard to be formalized, hence the need to propose tools for measuring, evaluating, and predicting cultural behavior [31]. To overcome such a challenge, Hofstede [35] defined the *Hofstede's 6-D framework*. It is a set of six dimensions that assume values from zero to one hundred and which combination characterizes a specific country globally [34, 35, 37].

In information systems research, Hofstede's dimensions are widely utilized [1, 11, 17]. For instance, Borchers et al. [11] investigated the impact of cultural factors on software engineering processes, analyzing three distinct cultures, i.e., Japan, India, and United States. Results showed how different cultures had different approaches to the software engineering process, e.g., Japanese developers are characterized by a high level of uncertainty avoidance, slowing down the decision-making process.

Despite the wide diffusion of the framework, several studies raised serious concerns about it [2, 8, 54], even supporting the rejection of its use [12, 13, 57]. For instance, Brewer and Venaik [12, 13] doubt the capacity of the tool in representing cultural profile. However, Venkateswaran and Ojha [69] showed how Hofstede's framework represents the most efficacious way to describe the complex world of cultural attitudes. In particular, they showed [69] how Hofstede's framework is characterized by a pragmatic approach, which has been shown effective in several fields of application [1, 11, 17], e.g., management, law, politics, ethics, architecture, medicine, economics, and computer science [36].

The six Hofstede's dimensions are defined as follows:

Power Distance Index. *PDI* expresses the degree to which the less powerful members of a society accept and expect that power is distributed unequally. People in societies exhibiting a high level of *Power Distance* accept a hierarchical order in which everybody has a determined place. On the contrary, in societies with low *Power Distance*, people strive to equalize the distribution of power and demand justification for power inequalities [35, 37].

Individualism vs. Collectivism. *IDV* represents the degree to which people in a society are integrated into groups. A high level of such a dimension indicates a society in which individuals are expected to take care of only themselves and their immediate families. Conversely, a low level indicates a preference for a tightly-knit framework in society in which individuals can expect their relatives or members of a particular group to look after them in exchange for unquestioning loyalty [35, 37].

Masculinity vs. Femininity. *MAS* represents a contrast between two preferences. The Masculinity side (high level) is defined as "a preference in society for achievement, heroism, assertiveness and material rewards for success". In contrast, the Femininity side (low level) represents "a preference for cooperation, caring for the weak and quality of life" [35, 37].

Uncertainty Avoidance. *UAI* expresses the degree to which the members of a society feel uncomfortable with uncertainty and ambiguity. Countries exhibiting high level of UAI maintain rigid codes of belief and behavior and are intolerant of unorthodox behavior and ideas. Conversely, a low level of UAI indicates societies that maintain a more relaxed attitude in which practice counts more than principles [35, 37].

Long vs. Short Term Orientation. *LTO* measures how much people are oriented toward a long-term outlook in contrast to a more short-term. A high degree in this index (Long-Term) indicates that people encourage thrift and efforts in modern education as a way to prepare for the future. On the contrary, a lower degree of this index (Short-Term) indicates that people tend to honor traditions and value steadfastness [35, 37].

Indulgence vs. Restraint. *IVR* refers to the degree of freedom that societal norms give citizens to fulfill their human desires. A high level (Indulgence) indicates a society that allows relatively free gratification of basic and natural human desires related to enjoying life and having fun. Conversely, a low level (Restraint) indicates a society that controls gratification of needs and regulates it using strict social norms [35, 37].

2.2 Community Smells

On the social and human perspective in Software Engineering (SE), several studies focused on *community smells*, i.e., sub-optimal patterns across the organizational and social structure in a software development community that are precursors of alarming and unforeseen socio-technical events. To better help in the understanding of such "smell", we reported, as an example, the one known as *Lone Wolf*. We have a *Lone Wolf* effect when, within a development community, there are unsanctioned or defiant contributors who carry out their work irrespective or regardless of their peers.

After their definition, the software engineering community started focusing on community smells only after the release of the tool CodeFace4Smells by Tamburri et al. [64]—an augmented version of CodeFace by Joblin et al. [39]—capable of automatically detecting four community, i.e., *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*. Through the use of repository mining and mailing lists analysis, such a tool can construct the so-called *developer networks*—namely, graphs describing the relations of collaboration and communication among developers—and using them identify software communities automatically.

Regarding the impact of community smells on software products, Palomba et al. [47] showed that such smells are among the top factors influencing the emergence of code smells—"poor implementation choices applied by developers during software evolution that often lead to critical flaws or failure" [30]—in source code. Along the same line, other researchers focused on establishing the impact of community smells on other dimensions of software engineering, e.g., architecture debt [44] and organizational structure types [65]. Indeed, Tamburri et al. [64] conducted a large-scale empirical study on 60 open-source ecosystems to evaluate the diffuseness of community smells, how they are perceived and how smells relate to existing socio-technical factors. In terms of diffusion, their results showed that community smells are common in open-source teams and perceived by developers as relevant problems for the evolution and sustainability of software communities.

Based on the results above, Palomba and Tamburri [48] provided the first machine learning approach able to predict community smells considering socio-technical metrics, obtaining promising results, i.e., F-Measure 78%. Furthermore, in the context of detecting such smells, Almarimi et al. [4] build a multi-label learning model based on genetic algorithms to detect eight common types of community smells. The tool evaluation involved 103 open-source projects and 407 smells instances and resulted better performance indexes compared to other solutions (F-measure of 89%).

More related to the refactoring and mitigation strategies for community smells, Catolino et al. [20] studied how socio-technical factors previously used influence the variability of community smells. Furthermore, they studied how developers remove such smells [23]: by surveying 76 experts, the authors were able to elicit and distill a set of refactoring operations generally applied by practitioners to remove the four community smell types identifiable through CodeFace4Smells. As an ulterior contribution, Catolino et al. [21] showed how the emergence of community smells might be potentially reduced by increasing gender diversity.

Despite other works studied factors able to influence the emergence of community smells [21, 44, 64, 65], they do not cover geographical and cultural related factors. Hence, we believe that an appropriate detection is needed to fill this gap. Our work aims to study the correlation between culture and geographical distribution of developers on the emergence of communication and collaboration problems, represented using community smells. This contribution covers (1) the geographical gap using the developers' geographical distribution and (2) both the gaps (geographical and cultural) considering culture as an independent factor.

3 RESEARCH METHODOLOGY

The *goal* of the paper is to assess to what extent cultural and geographical dispersion in open-source distributed communities influence such communities' communication and collaboration activities with the *purpose* of giving a deeper understanding of how software teams communicate and cooperate when globally distributed. The *perspective* is that of project managers, who are interested in effectively allocating resources, meet the projects requirements, or managing/monitoring complex organizational structures. Figure 1 shows an overview of the methodology followed in order to address our research questions.

3.1 Hypothesis and Research Questions

State of the art has demonstrated that culture and geographical distribution in software development teams may impact productivity and communication activities [24, 27, 28, 56]. Since some types of community smells have been used to represent these types of problems [21], it is reasonable to expect that they too will be impacted by geographical factors. The working hypothesis behind this work is the following:

The presence inside development team of developers from different cultures can impact the collaboration and communication activities, leading to the emergence of community smells.

In order to verify our working hypothesis, we formulate two research questions. The first aims to provide an overview of the cultural and geographical characteristics of open-source development communities, studying to what extent such communities are composed of developers from different cultures and places around the world. The primary intent is to verify that open-source communities are dispersed enough to justify a deeper study on the correlation between dispersions metrics and community smells.

> **RQ$_1$** To what extent are open-source communities culturally and geographically dispersed?

Open source developers tend to be aware of their team mates' countries of residence [67]. Hence, in the second research question, through repository mining and the use of linear regression modeling, the cultural and geographical dispersion—represented using the six Hofstede's dimensions—are correlated with a number of community smells, e.g., *Lone Wolf* and *Organizational Silo*.

> **RQ$_2$** To what extent do cultural and geographical dispersion within teams influence the number of community smells?

In terms of reporting, we employ the guidelines by Wohlin et al. [71] and we follow the *ACM/SIGSOFT Empirical Standards.*[1]

[1] Available at: https://github.com/acmsigsoft/EmpiricalStandards. Given the nature of our study and the currently available standards, we follow the *"General Standard"* and *"Data Science"* definitions and guidelines.

3.2 Context of Study

The context of the study consist of (1) open-source development communities, (2) cultural and geographical dispersion metrics, and (3) community smells. For sake of readability, we divide the section into three parts, one for each item above.

3.2.1 Open-Source Communities.

To conduct our study, we start from the dataset made available by Vasilescu et al. [68], which reports on 23 493 projects and corresponding communities on GitHub. Specifically, the dataset contains various time windows for each project, each distant 90-days from the previous one. This statement means that each row of the dataset had information about the community working on a specific project in a specific temporal window, e.g., number of committers and commits, number of women, and various socio-technical metrics. The most useful information for our study is the developers' country of origin, retrieved using the GitHub APIs. It is important to note that the GitHub APIs use the location descriptions on developers' profile pages, which are free-text optional entries. For that reason, the retrived locations include unstructured and often noisy data, for example, latitudes/longitudes, postcodes, IP-addresses, and fictitious addresses (e.g., "the most distant place from the center of the universe"). Therefore, after data preprocessing, only a part of the developers in the dataset have associated a "credible" country, that is, 39 102 (32.04%) of all the users in the entire dataset (122 014).

3.2.2 Dispersions metrics.

Our study focuses on cultural and geographical dispersion.

Cultural Dispersion. The *cultural dispersion* of a development community is the degree in which developers form a community with different cultural attitudes [65]. A low level indicates that the community members come from the same cultural education; the opposite, a high level indicates that the community members present different cultural attitudes. To operationalize the cultural dispersion of a community we have used the Hofstede's 6-D model (described in Section 2.1). Specifically, we have chosen to represent cultural dispersion using six metrics, one for each Hofstede dimension. Each of these metrics corresponds to the standard deviation of the set containing the community members' values for one of the six dimensions, and can assume values from zero to fifty.

Geographical Dispersion. The *geographical dispersion* of a development community is the degree to which its members work from different and distributed places in the world. A low level indicates that the community members work from the same or near locations; the opposite, a high level suggests that the community members work from different and distant places. In our case, to operationalize the geographical dispersion of a development community we have used the spherical distance between its members. Specifically, we have computed the standard deviation of the set of physical distances between each community member, which is expressed as the spherical distance between their locations. From a practical point of view, we use the Python library GeoPy[2] to calculate the spherical distance (in miles) from one address to another. In case we do not have the precise position of the developers, we use the center point of the most specific place in the address string.

[2] https://geopy.readthedocs.io/en/stable/index.html

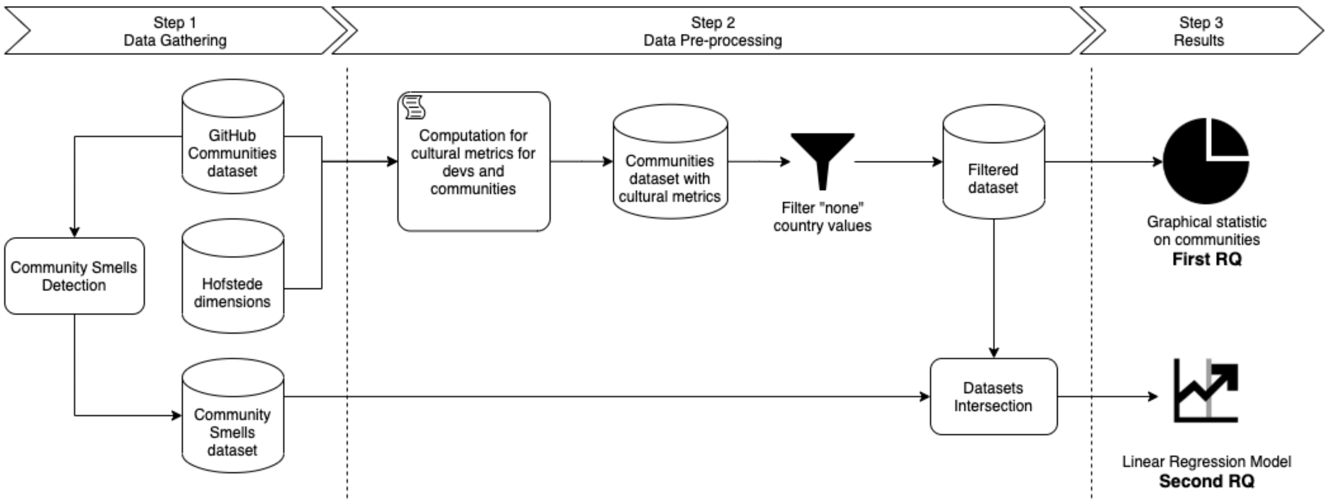

Figure 1: Overview of our research methodology.

3.2.3 Community Smells. For our study, we have used four types of community smells—i.e., *Organizational Silo*, *Black Cloud*, *Lone Wolf*, and *Radio Silence*—that have been used in previous studies to represents communication and collaboration problems [21, 60]. The four types of community smells are outlined as follows:

Organizational Silo Effect. This form of social debt refers to the presence of siloed areas of the community that do not communicate, except through one or two of their respective members;

Black Cloud Effect. This community smell reflects an information overload due to lack of structured communications or cooperation governance;

Lone Wolf Effect. This smell arises when the development community presents unsanctioned or defiant contributors who carry out their work with little consideration of their peers, their decisions and communication;

Radio Silence Effect. The smell appears when one member interposes herself into every formal interaction across more subcommunities with little flexibility to introduce other channels.

Detailed explanation of each community smell is available in the original papers [21, 62].

3.3 Data Collection

3.3.1 Data Collection for RQ_1. To address RQ_1, we compute cultural and geographical information—described in Section 3.2.2—of the developers GitHub, performing the following steps for each development community in the Vasilescu et al. [68] dataset:

(1) We extract the origin country of each developer;
(2) We compute the geographical dispersion as the standard deviation of the set of physical distances between each community member, which is expressed as the spherical distance between their working locations—computed using GeoPy.
(3) We obtain Hofstede's dimensions values for each country of each developers;
(4) We compute the six cultural dispersions, each one as the standard deviation of the set containing the community members' values for one of the six dimensions.

To sum up, the dataset for this research question consists of the dataset of Vasilescu et al. [68] augmented with the dispersions metrics computed using the developers' origin country.

3.3.2 Data Collection for RQ_2. For RQ_2, we need to study the correlation between cultural and geographical dispersion and the number of instances of four types of community smells, i.e., *Organizational Silo*, *Black Cloud*, *Lone Wolf*, and *Radio Silence*.

We have decided to focus on these specific smells since they are used by the research community [20, 21, 23] for formalizing communication and collaboration problems. Information about these smells instances, are already available in the dataset of Catolino et al. [21]. In particular, starting from the dataset of Vasilescu et al. [68], Catolino et al. randomly extracted 25 projects and computed community smells using the CodeFace4Smells tool [64].

To sum up, the dataset for this research question consists of the dataset of Catolino et al. [21]—containing community smells metrics—intersected with the dataset used for the first research question—containing cultural and geographical metrics. In this way, we obtain a dataset that, for each community in the Catolino et al. [21] one, combines both community smells and cultural metrics.

3.4 RQ_1 - Culture and Geography of Open-Source Communities

Starting from the dataset of Vasilescu et al. [68], we compute the values, average, and standard deviation for each Hofstede dimension per development community. Furthermore, we also computed the geographical dispersion for each community, as described in Section 3.3. Then, we create a dataset combining the one from Vasilescu and the cultural and geographical measures mentioned above. Consequently, we use graphical representations to show to what extent the communities are (1) geographically dispersed and

(2) culturally dispersed internally. We choose to represent only the values corresponding to the communities for which we know more than 50% of the members' countries (500 projects, with multiple time window). We intend to take only the communities for which the computed cultural characteristics are the most accurate.

3.5 RQ$_2$ - Building a Statistical Model

To address **RQ$_2$**, we define a statistical linear regression model relating a development community's cultural and geographical dispersion to the frequency of community smells in the same group.

In the following subsections, we explain the variables used to build our model. All the metrics presented in the following have been calculated for each time window since that the community's members of a project change during time.

3.5.1 Independent Variables. Based on our hypothesis, we consider seven factors as independent variables.

The first six represent **the cultural dispersion of a development community**: the rank in which developers form a community with different cultural attitudes (measured using the Hofstede 6-D model—described in Section 2.1). As in Section 3.4, to operationalize the cultural dispersion of a community, we have chosen to use six values, one for each Hofstede dimension. Each of these values corresponds to the standard deviation of the set containing the community members' values for one of the six dimensions.

The last factor is the **Geographical Dispersion** defined as the standard deviation of the set of physical distances between each community member—calculated using the origin country coordinates already present in the dataset of Vasilescu et al. [68].

3.5.2 Response Variables. We aim at understanding the impact of cultural and geographical dispersion on the presence of community smells mentioned in Section 3.3.2, i.e., *Radio Silence*, *Lone Wolf*, *Black Cloud*, and *Organizational Silo*. For that reason, our four response variables are **the four numbers of instances for each community smell**.

3.5.3 Control Variables. While our hypothesis aims at investigating the relationship between cultural/geographical dispersion and the presence of community smells, other community-related factors might influence the response variable, as demonstrated in literature [20, 21, 49]. More particularly, all the following factors have a strong correlation with the emergence of community smells:

- **Number of Committers**: It is the number of people that have done at least one commit in a given project time window. Catolino et al. [21] have demonstrated that the number of committers in a community can influence the number of community smells. Therefore, we considered the total committer count of a project as the first control variable.
- **Number of Commits**: It is the total commit count in a project. In the majority of the case, a high number of commits may indicate a high activity in the community. Since that it is not rare that more developers work on the same code module or functional requirement, such an activity might be corresponding with high communication and collaboration, possibly impacting the number of community smells.
- **Team Size**: It is the number of contributors per team in a given temporal window. The community's size can influence the number of community smells it contains, for example, incrementing the number of smells with an increment of community members.
- **Turnover**: It is the fraction of the team in a given temporal slice that is different with respect to previous windows (i.e., the *turnover ratio*). A high turnover indicates that community members changed frequently. The constant introduction of new contributors might lead to communication problems.
- **Project Age**: It is the difference between the maximum index and the index of the 90-day temporal interval during which the first commit was recorded. Older projects and their teams may have experienced different trends or work habits, and these changes might affect the presence of smells.
- **Tenure diversity**: *Tenure measures* have been used in state of the art to represents the experience of developers in various fields [43], showing how they are able to influence the emergence of community smells [21]. In our dataset, we considered two typologies of tenures: (1) *commit tenure* (that measures the coding experience of a contributor within all GitHub projects in which he has contributed) and (2) *project tenure* (that measures the developer experience in the specific project considered).
- **Tenure median**: Remaining in the field of tenure metrics, to represent an average of project and commit tenures for what concern developers of a community, also the project and commit median tenure has been considered.
- **Number of women in a team**: The number of women is computed as the difference between the total number of community members and the number of men belonging to the community. Catolino et al. [21] demonstrate that number of women can influence the occurrences of some community smells, that is, *Black Cloud* and *Radio Silence*.
- **Blau-Index**: Blau [10] defined *Blau diversity index* as $1 - \sum_{i=1}^{n} P_i^2$ where P_i refers to the percentage of female team members. The values fluctuate between 0 and 0.5, at which there is the same percentage of male and female board members and thus the diversity is maximized.
- **Socio-Technical Congruence (STC)**: As defined by Valetto et al. [66], *STC* represents "the state in which a software development organization harbors sufficient coordination capabilities to meet the coordination demands of the technical products under development" [66]. Catolino et al. [21] have operationalized it as the number of development collaborations that do communicate over the total number of collaboration links present in the collaboration network.
- **Truck Factor (TF)**: It is the minimum number of member of a community that have to quit (or being hit by a bus) before the project will fail [6, 7, 29, 70]. In their dataset, Catolino et al. [21] operationalized truck factor based on core and peripheral community structures identified by CodeFace4Smells [64], as the degree of ability of the development community to remain connected without its core part.
- **Centrality**: It is the strength of a community and it is based on modularity measures [32]. A value over 0.3 indicates that the community is highly modular and thus with a clear distinction of the sub-communities present in its development network. Opposite, a value below 0.3 indicates that there are no sub-communities.

The above mentioned factors were considered as control variables in our statistical models.

3.5.4 Statistical Model Construction. The response variables consist of the number of instances of four types of community smells: *Organizational Silo*, *Black Cloud*, *Lone Wolf*, and *Radio Silence*. Therefore, we build four statistical models, i.e., one for each community smell considered, all with the same independent and control variables. Since the dataset provided in [21] consisted of multiple temporal windows for each project—following also the experiment performed by Catolino et al. [21]—we build four *linear mixed models* to capture measurements from within the same group (i.e., within the same project) as a random effect [42]. In our case, we use the time window as a random effect and all other variables as fixed effects. To build our model we use the functions `lmer` and `lmer.test` available in the `R` package `lme4` [9]. In order to avoid the problem of multi-collinearity [46], we employ a stepwise variable removal procedure based on the *Companion Applied Regression* (`car`) `R` package,[3] and in particular based on the `vif` function [46]. Finally, the effect sizes of the coefficients are obtained using the ANOVA statistical test [25] and are considered important if they were statistically significant (that is, the p-value is less than 0.05).

To better analyze and interpret our findings, we build other two baseline statistical models: the first one containing all the control factors and the random effect, but not the independent factors; the second one containing only the random factor. After that, we compared the models with the corresponding baselines using the *AIC* (*Akaike information criterion*) and *BIC* (*Bayesian information criterion*) [3, 15]. We choose these index since they are used as model selection criteria [16], indeed, in our case we had three different models to evaluate. *AIC* and *BIC* are estimators of prediction error and quality of statistical models for a given data set. These metrics estimate the quality of each model in order to provide a means for model selection [3]. The general rule is that the model with the lower AIC and BIC is the one that better characterizes the sample. The comparison with the first model allows us to study how the control variables, without the independent factors, influence the number of community smells. The comparison with the second model allows us to understand whether the obtained results reflected the random effect.

4 ANALYSIS OF THE RESULTS

This section illustrates the results for each research question.

4.1 RQ$_1$ - Culture and Geography of Open-Source Communities

As we have said in Section 3.4, we have studied to what extent open-source communities, on GitHub, are (1) geographically dispersed and (2) culturally dispersed internally.

Figure 2 reports the geographical dispersion values for each community in the dataset of Vasilescu et al. [68]. We choose to use violin plots to represent also the dispersion and the differences between the various communities. As shown, most values are near zero, namely, no geographical dispersion. However, another great set contains communities with a value of nearly three thousand, and the median and the mean are near the two thousand value.

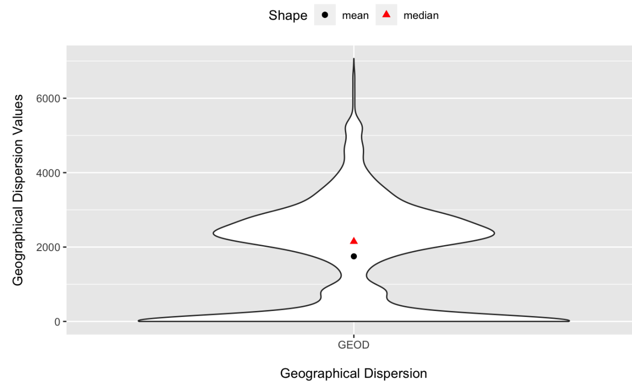

Figure 2: Geographical dispersion values for communities.

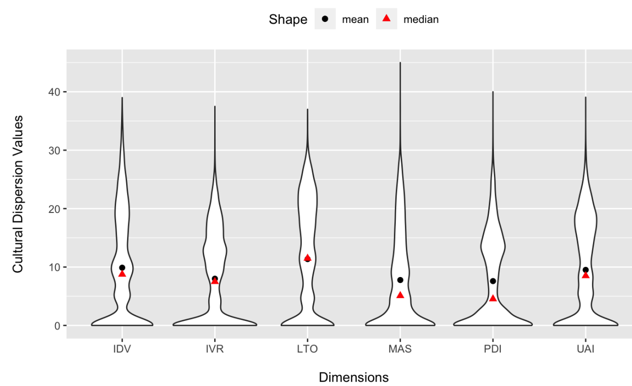

Figure 3: Cultural dispersion values for communities.

Figure 3 reports the standard deviation of each community per Hofstede dimension. We have used such statistics to operationalize cultural dispersion in development communities. We note that the majority of the communities present zero (or near zero) cultural dispersion (as also represented by the fact that the means and medians of all violin plots are near zero value). In general, it is possible to affirm that most of the values are contained in the interval from zero and twenty-five. The dimension with more dispersed values is *Long Term Orientation* (LTO), as shown by the median and mean values greater compared to the other dimensions.

RQ$_1$: summary of the results.

☞ Our statistical analysis revealed that GitHub presents communities with both low and high geographical dispersion. Concerning cultural dispersion, our work reveal an unexpected result: GitHub is not extremely culturally dispersed as one could expect. In any case, we found valuable differences to conduct a deeper study on the relationship between such dispersions and the number of smells.

[3] https://cran.r-project.org/web/packages/car/index.html

4.2 RQ_2 - Building a Statistical Model

In this section, we report the results that address the $\mathbf{RQ_2}$. We built four mixed linear models, one for each of the community smells types considered, to study the relation between cultural and geographical metrics—plus others control factors—and the number of smells in a software development community.

In the following sections, we discuss the obtained results separately for each type of the considered smells.

Table 1: Results achieved for *Radio Silence.*

Factor	All Variables		Conf. Variables		Random
	Estimate	Sig.	Estimate	Sig.	Estimate
(Intercept)	-4.042		-1.926		2.261
# committers	-0.091		-0.093		
# commits	0.101		0.024		
projectAge	-0.015		-0.027	.	
turnover	9.857	***	9.982	***	
blauGender	4.011	**	0.492		
tenureMedian	0.067	*	0.049		
tenureDiv	0.008		0.011		
teamSize	0.304	.			
stCongruence	-0.336	.	-0.391	*	
truckFactor	-0.018		-0.012		
# females	0.001		0.042	***	
expertise	0.066		0.054		
centrality	-0.118		-0.074		
PDID	0.071	*			
IDVD	-0.055	*			
MASD	-0.001				
UAID	-0.007				
LTOD	0.036				
IVRD	-0.013				
GeoD	-0.001	*			

***:p<0.001; **:p<0.01; *:p<0.05; .:p<0.1

4.2.1 Radio Silence. Table 1 reports the details regarding the statistical model built for the *Radio Silence* effect. As we can see, software community members' *turnover* is a potent estimator over the occurrence of the target community smell. The second most important factor is the *Blau Diversity Index*, indicating that the presence of women in development teams can influence the number of smells.

As regards the independent factors, *Power Distance Index Dispersion* impacts the response variable with an estimate of 0.07 and a standard error of 0.031 with significance below 0.013. Moreover, also *Individualism vs Collectivism Dispersion* impacts the response variable, with an estimate of -0.052 and a standard error of 0.021 with significance below 0.018. Finally, *Geographical Dispersion* impacts the response variable with an estimate of 0.012 and a standard error of 0.001 with significance below 0.043.

Regarding the comparison with the baseline, our model achieved AIC and BIC values (1) equal to the one with the confounding variables and (2) less than the one with only the random factor. This means that the addition of the independent factors does not necessarily help in better explaining the response variable.

4.2.2 Organizational Silo. Table 2 reports the details regarding the statistical model built for the *Organizational Silo* effect. As we have also seen in the previous section, *Turnover* is an influential factor in the emergence of the smell. Also, the total *number of committers* influence the number of smells significantly. We can explain the two results mentioned above because a high number of committers combined with an exceedingly static community (that is, that does not change very much) can lead to a division in consolidated sub-communities.

Table 2: Results achieved for *Organizational Silo.*

Factor	All Variables		Conf. Variables		Random
	Estimate	Sig.	Estimate	Sig.	Estimate
(Intercept)	4.997		2.419		2.286
# committers	0.402	**	0.423	**	
# commits	-0.218	.	-0.215	.	
projectAge	-0.024		-0.033		
turnover	-1.601	**	-1.738	**	
blauGender	0.414		2.514		
tenureMedian	-0.066		-0.057		
tenureDiv	0.039		0.028		
teamSize	0.133				
stCongruence	0.691	*	0.531		
truckFactor	0.025		0.046		
# females	0.017		0.014		
expertise	-0.469		-0.462		
centrality	0.086		-0.084		
PDID	0.001				
IDVD	0.008				
MASD	-0.033				
UAID	0.013				
LTOD	-0.023				
IVRD	-0.023				
GeoD	-0.001	***			

***:p<0.001; **:p<0.01; *:p<0.05; .:p<0.1

Geographical Dispersion impact (negatively) the number of *Organizational Silo*, with an estimate of -0.001 and a standard error below 0.001 with significance below 0.001.

Regarding the comparison with the baseline, our model achieved AIC and BIC values (1) slightly lower than the ones with the confounding variables and (2) equal to those with only the random factor. This means that the addition of the independent factors could help in better explaining the response variable, but that deeper analysis is needed to exclude the influence of the random factor.

4.2.3 Lone Wolf. Table 3 reports the details regarding the statistical model built for the *Lone Wolf* effect. For this type of smell, *team size* and *socio-technical congruence* seem to be the most relevant factors. Regarding the cultural aspects, *Individualism vs Collectivism Dispersion* impacts the number of smells, with an estimate of -0.194 and a standard error of 0.083 with significance below 0.05. This result can indicate that poorly coordinated teams tend to exhibit more *Lone Wolf* effect instances, but that team members with different ideas about the concepts of individualism and collectivism can mitigate their emergence.

Regarding the comparison with the baseline, our model achieved AIC and BIC values (1) lower than those with the confounding variables and (2) significantly lower than those with only the random factor. Thus, we can assert that the addition of the independent factors can help in better explaining the response variable.

4.2.4 Black Cloud. Table 4 reports the details regarding the model built for the *Black Cloud* effect. As shown, *Blau-index*, which denotes diversity, has a firm estimate and significance.

Table 3: Results achieved for *Lone Wolf*.

Factor	All Variables		Conf. Variables		Random
	Estimate	Sig.	Estimate	Sig.	Estimate
(Intercept)	-9.123		8.756		8.188
# committers	0.361		0.226		
# commits	0.216		0.118		
projectAge	0.093	.	0.123	*	
turnover	0.724		1.285		
blauGender	-4.044		-0.933		
tenureMedian	-0.052		-0.008		
tenureDiv	0.045		0.065		
teamSize	4.777	***			
stCongruence	-7.654	***	-14.092	***	
truckFactor	-0.221		-0.201		
# females	0.004		0.016		
expertise	0.784		0.661		
centrality	0.078		0.958	.	
PDID	-0.162				
IDVD	-0.194	*			
MASD	0.001				
UAID	0.158				
LTOD	0.047				
IVRD	0.094				
GeoD	-0.001				

***:p<0.001; **:p<0.01; *:p<0.05; .:p<0.1

Table 4: Results achieved for *Black Cloud*.

Factor	All Variables		Conf. Variables		Random
	Estimate	Sig.	Estimate	Sig.	Estimate
(Intercept)	7.266		5.923		4.466
# committers	-0.011		-0.046		
# commits	-0.018		0.017		
projectAge	-0.005		-0.017		
turnover	0.785		1.127		
blauGender	-7.668	**	-10.906	***	
tenureMedian	0.042		0.025		
tenureDiv	-0.001		-0.005		
teamSize	0.171				
stCongruence	0.249		0.003		
truckFactor	0.028		-0.041		
# females	0.017		0.035		
expertise	-0.249		-0.331		
centrality	-0.276		-0.482		
PDID	-0.004				
IDVD	-0.021				
MASD	0.071				
UAID	0.035				
LTOD	-0.016				
IVRD	0.009				
GeoD	-0.001	***			

***:p<0.001; **:p<0.01; *:p<0.05; .:p<0.1

Geographical Dispersion impacts the number of smells, with a estimate of -0.001 and a standard error of 0.001 with significance below 0.001. Thus we conclude that having a diverse team in terms of gender and geographical dislocation help reducing the number of *Black Cloud* instances. We can explain this by the fact that having dislocated team members leads to the necessity of introducing structural communication protocols in a community environment.

Regarding the comparison with the baseline, our model achieved: (1) an AIC value lower than the one with the confounding variables and the one with only the random factor and (2) a BIC value equal to the one with the confounding variables and greater than the one with only the random factor. Then, we can assert that the addition of the independent factors could help explain the response variable better, but that deeper analysis is needed.

RQ$_2$: summary of the results.

☞ *Cultural Dispersion* is a relevant factor for the emergence of *Radio Silence* and *Lone Wolf*, while *Geographical Dispersion* is relevant for all the smells except *Lone Wolf*. Nonetheless, further research on the impact of cultural and geographical dispersion with *Radio Silence* would be desirable.

5 THREATS TO VALIDITY

This section illustrates the threats to the validity of the study and the way we mitigated them.

Threats to Construct Validity. Threats to construct validity concern the relation between theory and observation and are mainly due to imprecision in performed measurements. The main threat is related to the representation used for defining and measuring culture. In order to characterize culture and cultural dispersion, we used Hofstede's framework (also known as Hofstede's 6-D model). Although, a few researchers raised concerns about this framework [2, 8, 54], Venkateswaran and Ojha showed how the framework is the most efficient way to represent the complex world of cultural attitudes. The second threat regards the sources used for conducting our statystcal analysis. Raw information about cultural and geographic factors comes from a public dataset provided by Vasilescu et al. [68], while information about community smells and control variables belong to the dataset of Catolino et al. [21]. Both the dataset are well known and used by research community. Of course, we cannot exclude possible inaccuracies in the computation of variables, for this reason we double checked that everything looked fine before conducting our experiments. Finally, the use of standard deviation could introduce some threats cause that it is unreliable in case of skewed measures. To address this, we use statistical tests to verify the normality of the data before conducting our studies.

Threats to Conclusion Validity. Threats to *conclusion* validity concern the relation between treatment and outcome and are related to issues that affect the ability to draw the correct conclusions at the end of the work. Most of these threats regard the statistical models built for the last research question; in particular, there was the possibility of omitting some variables that could influence the phenomenon of community smells. Therefore, a first mitigation strategy consisted of using all the control factors identified by previous literature that showed correlation with community smells [20, 21, 49]. Last, we investigated the influence of the aforementioned variables building two baselines models used for evaluating the factors' impact on the response variables, shown in Section 3.5. Furthermore, as recommended by previous literature on the topic [20, 21], we used *mixed-effects model* [9, 42] for dealing with multiple time windows for each project. This choice allows us to capture measurements from within the same group. Moreover, we took into account the problem of multicollinearity [46] that occurs when independent variables in a regression model are correlated.

Indeed, we used the *variance inflation factors* [46] that detects multicollinearity in regression analysis, using R tool. In addition, we discarded outliers [46] and used only data for which the information on the origin country was available for most developers. Finally, in order to verify the significance of the independent variables on the response variable, we employed ANOVA [25] test, which is well known as an useful method to evaluate the results of models.

Threats to External Validity. Threats to *external* validity are conditions that limit the ability to generalize the results of our experiment to the real world. The riskiest threat corresponds to the extremely high number of missed values for the origin country of developers, that could lead to an incorrect cultural characterization. To mitigate this aspect, we used only communities with a few missed countries during our experiment. However, we plan further actions to fix missed information to continue our research. Furthermore, we could not exclude some imprecision regarding the origin country of the developers, see that it was computed using GitHub developers' profile page information. In order to mitigate that, we performed a large-scale empirical study on many developers to increase the possibility of having many correct and precise data.

6 DISCUSSION AND IMPLICATIONS

The modern state of the art has not yet investigated the role of software team members' culture as a factor for the emergence of community smells. In this work, we aimed to add shed light over cultural impacts over known and nasty phenomena across software organisational structures through quantitative empirical researches.

6.1 Statistical Analysis: an Overview

Focusing on the statistical overview, we reported several interesting findings with ramifications beyond the subject matter. First, our analysis showed that US developers dominate open-source, as already demonstrated by previous works [52, 59]. The finding here is that a US-cultured approach is bound to be predominant when it comes to global software engineering and therefore specific care may be needed. Second, concerning geographical dispersion, we reported communities with both low and medium values. The proposition here is that people from different parts of the world tend to congregate in communities whose center is physically very far from them. Lastly, concerning cultural dispersion, open-source communities are not as highly culturally dispersed as we could expect. Such a result is interesting, seeing that one of the main aspects of GitHub is to allow people from different places around the world to work together. Conversely, such a result may have been influenced by the large amount of US developers in the dataset. However, such a high number also indicates that the various countries actively involved in open-source tend to have similar ideas to the aforementioned US-dominant logic, a pattern consistent with the phenomenon known as *tunnel-vision* [26]; this phenomenon may require additional investigation.

6.2 Community Smells in Global Contexts: Considerations and Lessons Learned

Focusing on the cultural and geographical dispersion as mediators for community smells, we reported diverse results for each smell.

Black Cloud and Organizational Silo. First, in the context of *Black Cloud* and *Organizational Silo*, geographically dispersed team members can influence negatively the presence of the smells. For the Black Cloud, a possible motivation is that managing people physically arranged in different parts of the world means using specific management tools and protocols for communication and collaboration, e.g., *Trello* and *Jira* which "nudge" the way of working towards a rather narrow, more disciplined approach. Conversely, one can consider this fact a critical problem because it is a way to influence the organisational structure—by usage of specific tools—rather than a deliberate choice of the individuals. In summary, the imposition exerted in the aforementioned case may generate community smells that we are yet to discover and evaluate. Similarly, for *Organizational Silo*, having team members located nearby can lead to the formation of sub-communities. Conversely, one can consider such a strategy as an indirect—and again "forced"—way to stress on standard practices, e.g., avoiding group-think in sub-communities.

The only conclusion that can be drawn is that global settings around community smells make their management even more into a balancing act which would be better instrumented with proper automated tooling.

Radio Silence. More specifically, for cultural dispersion, we have seen that *Power Distance Index Dispersion* influences positively the emergence of *Radio Silence* effects. An interpretation of this is connected to communities that agree on the assignment of power to individual members and which are led to leave the communication tasks between the various sub-communities to such individuals.

Lone Wolf. Lone wolves reflect rather individualist ways of working and therefore bring on the table the *Individualism vs. Collectivism and Dispersion*, an essential trade-off for the emergence of *Lone Wolf* effects in the first place. Considering that US culture tends to promote individualism a low level of *Individualism vs. Collectivism Dispersion* may reflect an acceptance of individualistic attitudes, which leads to the emergence of *Lone Wolf* instances. This phenomenon and its occurrences deserve further research to ascertain its precursors and effects.

Control Variables. Our study confirms previous findings [20, 21]: indeed, socio-technical [19, 66], e.g., *truck factor*, and diversity metrics [21], i.e., *Blau-index*, are strongly correlated with the emergence community smells. From here, the need to possibly starting investigating causality relationship and profiling better each of this metric providing new practical insights, e.g., threshold related to level of warning.

Looking for Inclusiveness Patterns. Last, the results of this paper possibly shed light on possible issues when working in a dynamic environment in terms of diversity, so these need to be better contextualized and studied. Formalizing these issues proposing inclusiveness patterns represents our next step in our future schedule.

7 CONCLUSIONS

This article reports on empirical evidence to clarify the connection between *cultural and geographical dispersion* and problems in communication and collaboration activities of software development teams. Our study reveals that the two types of dispersion impact the emergence of all community smells, yet not necessarily in a negative fashion: in this sense we conclude that there are not only negative *"fences"* to look at. The strongest result consists of the correlation between the presence of individualistic and collectivist people and the emergence of *Lone Wolf* effects.

In terms of contributions, our work provides three:

(1) The first, large-scale empirical exploration of the cultural dispersion of 23 493 communities on the GitHub platform using Hofstede's 6-D framework [35];
(2) Four regression models that analyze the influence of cultural and geographical on the emergence of four types of community smells, e.g., *Lone Wolf* on open-source communities;
(3) A publicly available replication package [41] supporting further investigation on the topic and containing all the findings.

As a future agenda, we plan to conduct qualitative studies—e.g., surveys, interviews, and focus groups—with the aim of enhancing the generalizability of the obtained results.

ACKNOWLEDGMENTS

Gemma is partially supported by the European Commission grant no. 825040. Fabio is partially supported by the Swiss National Science Foundation - SNF Project No. PZ00P2_186090.

REFERENCES

[1] Sameer Abufardeh and Kenneth Magel. 2010. The impact of global software cultural and linguistic aspects on Global Software Development process (GSD): Issues and challenges. In *4th International Conference on New Trends in Information Science and Service Science*. 133–138.
[2] Galit Ailon. 2008. Mirror, mirror on the wall: Culture's consequences in a value test of its own design. *Academy of management review* 33, 4 (2008), 885–904.
[3] Hirotogu Akaike. 1998. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*. Springer, 199–213.
[4] Nuri Almarimi, Ali Ouni, Moataz Chouchen, Islem Saidani, and Mohamed Wiem Mkaouer. 2020. On the detection of community smells using genetic programming-based ensemble classifier chain. In *Proceedings of the 15th International Conference on Global Software Engineering*. 43–54.
[5] Nuri Almarimi, Ali Ouni, and Mohamed Wiem Mkaouer. 2020. Learning to detect community smells in open source software projects. *Knowledge-Based Systems* 204 (2020), 106201.
[6] Guilherme Avelino, Eleni Constantinou, Marco Tulio Valente, and Alexander Serebrenik. 2019. On the abandonment and survival of open source projects: An empirical investigation. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2019, Porto de Galinhas, Recife, Brazil, September 19-20, 2019*. IEEE, 1–12. https://doi.org/10.1109/ESEM.2019.8870181
[7] Guilherme Avelino, Leonardo Passos, Andre Hora, and Marco Tulio Valente. 2016. A novel approach for estimating truck factors. In *2016 IEEE 24th International Conference on Program Comprehension (ICPC)*. IEEE, 1–10.
[8] Rachel F Baskerville. 2003. Hofstede never studied culture. *Accounting, organizations and society* 28, 1 (2003), 1–14.
[9] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2014. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823* (2014).
[10] Peter Michael Blau. 1977. *Inequality and heterogeneity: A primitive theory of social structure*. Vol. 7. Free Press New York.
[11] Greg Borchers. 2003. The software engineering impacts of cultural factors on multi-cultural software development teams. In *25th International Conference on Software Engineering, 2003. Proceedings*. IEEE, 540–545.
[12] Paul Brewer and Sunil Venaik. 2012. On the misuse of national culture dimensions. *International Marketing Review* (2012).
[13] Paul Brewer and Sunil Venaik. 2014. The ecological fallacy in national culture research. *Organization Studies* 35, 7 (2014), 1063–1086.
[14] Frederick P Brooks Jr. 1995. *The mythical man-month: essays on software engineering*. Pearson Education.
[15] Kenneth P Burnham and David R Anderson. 2004. Multimodel inference: understanding AIC and BIC in model selection. *Sociological methods & research* 33, 2 (2004), 261–304.
[16] Kenneth P Burnham and David R Anderson. 2004. Multimodel inference: understanding AIC and BIC in model selection. *Sociological methods & research* 33, 2 (2004), 261–304.
[17] Valentine Casey. 2011. Imparting the importance of culture to global software development. *ACM inroads* 1, 3 (2011), 51–57.
[18] Valentine Casey and Ita Richardson. 2008. A structured approach to global software development. *European systems and software process improvement and innovation, Dublin, Ireland* (2008).
[19] Marcelo Cataldo, Patrick A Wagstrom, James D Herbsleb, and Kathleen M Carley. 2006. Identification of coordination requirements: Implications for the design of collaboration and awareness tools. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. 353–362.
[20] Gemma Catolino, Fabio Palomba, Damian Andrew Tamburri, and Alexander Serebrenik. 2021. Understanding community smells variability: A statistical approach. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 77–86.
[21] Gemma Catolino, Fabio Palomba, Damian Andrew Tamburri, Alexander Serebrenik, and Filomena Ferrucci. 2019. Gender diversity and women in software teams: How do they affect community smells?. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, 11–20.
[22] Gemma Catolino, Fabio Palomba, Damian Andrew Tamburri, Alexander Serebrenik, and Filomena Ferrucci. 2020. Gender Diversity and Community Smells: Insights From the Trenches. *IEEE Softw.* 37, 1 (2020), 10–16. https://doi.org/10.1109/MS.2019.2944594
[23] Gemma Catolino, Fabio Palomba, Damian Andrew Tamburri, Alexander Serebrenik, and Filomena Ferrucci. 2020. Refactoring community smells in the wild: the practitioner's field manual. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society*. 25–34.
[24] Sébastien Cherry and Pierre N Robillard. 2004. Communication problems in global software development: Spotlight on a new field of investigation. In *International Workshop on Global Software Development, International Conference on Software Engineering, Edinburgh, Scotland*. IET, 48–52.
[25] Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. 2004. An anova test for functional data. *Computational statistics & data analysis* 47, 1 (2004), 111–122.
[26] Wendy L. Currie. 1999. Management Innovation and Change Panaceas: Strategic Vision or Tunnel Vision?. In *Proceedings of the Seventh European Conference on Information Systems, ECIS 1999, Copenhagen, Denmark, 1999*, Jan Pries-Heje, Claudio U. Ciborra, Karlheinz Kautz, Josep Valor, Ellen Christiaanse, David E. Avison, and Claus Heje (Eds.). Copenhagen Business School, 84–98.
[27] Sadhana Deshpande, Ita Richardson, Valentine Casey, and Sarah Beecham. 2010. Culture in Global Software Development - A Weakness or Strength?. In *2010 5th IEEE International Conference on Global Software Engineering*. 67–76. https://doi.org/10.1109/ICGSE.2010.16
[28] Christof Elbert. 2010. Global Software Engineering: Distributed Development, Outsourcing, and Supplier Management. *IEEE Computer Society Books* (2010).
[29] Mívian Ferreira, Marco Tulio Valente, and Kecia Ferreira. 2017. A comparison of three algorithms for computing truck factors. In *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*. IEEE, 207–217.
[30] Martin Fowler. 2018. *Refactoring: improving the design of existing code*. Addison-Wesley Professional.
[31] Adrian Furnham. 2012. *The psychology of behaviour at work: The individual in the organization*. Psychology press.
[32] John-Paul Hatala and Joseph George Lutta. 2009. Managing information sharing within an organizational setting: A social network perspective. *Performance Improvement Quarterly* 21, 4 (2009), 5–33.
[33] James D Herbsleb and Deependra Moitra. 2001. Global software development. *IEEE software* 18, 2 (2001), 16–20.
[34] Geert Hofstede. 2001. *Culture's consequences: Comparing values, behaviors, institutions and organizations across nations*. Sage publications.
[35] Geert Hofstede. 2011. Dimensionalizing cultures: The Hofstede model in context. *Online readings in psychology and culture* 2, 1 (2011), 2307–0919.
[36] Geert Hofstede. 2017. 50 years Memory lane – Developing cultural dimensions from IBM data. *Software of the Mind 2.0* (Sep 2017).
[37] Geert Hofstede, Gert Jan Hofstede, and Michael Minkov. 2005. *Cultures and organizations: Software of the mind*. Vol. 2. Mcgraw-hill New York.
[38] Project Management Institute. 2021. *A Guide to the Project Management Body of Knowledge* (7 ed.). 250 pages.
[39] Mitchell Joblin, Wolfgang Mauerer, Sven Apel, Janet Siegmund, and Dirk Riehle. 2015. From developer networks to verified communities: A fine-grained approach. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 563–573.
[40] Robert Kreitner, Angelo Kinicki, and Marc Buelens. 1999. *Organizational Behavior* (first european edition ed. london uk ed.). McGraw-Hill Publishing Company.
[41] Stefano Lambiase, Gemma Catolino, Damian Andrew Tamburri, Alexander Serebrenik, Fabio Palomba, and Filomena Ferrucci. 2021. Good Fences Make Good Neighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells — online appendix. https://doi.org/10.6084/m9.figshare.16802158

[42] Mary J Lindstrom and Douglas M Bates. 1988. Newton—Raphson and EM algorithms for linear mixed-effects models for repeated-measures data. *J. Amer. Statist. Assoc.* 83, 404 (1988), 1014–1022.

[43] Sarah MacCurtain, Patrick C Flood, Nagarajan Ramamoorthy, Michael A West, and Jeremy F Dawson. 2010. The top management team, reflexivity, knowledge sharing and new product performance: A study of the Irish software industry. *Creativity and Innovation Management* 19, 3 (2010), 219–232.

[44] Antonio Martini and Jan Bosch. 2017. Revealing social debt with the CAFFEA framework: An antidote to architectural debt. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. IEEE, 179–181.

[45] Audris Mockus and James Herbsleb. 2001. Challenges of global software development. In *Proceedings seventh international software metrics symposium*. IEEE, 182–184.

[46] Robert M O'brien. 2007. A caution regarding rules of thumb for variance inflation factors. *Quality & quantity* 41, 5 (2007), 673–690.

[47] Fabio Palomba, Damian Andrew Tamburri, Francesca Arcelli Fontana, Rocco Oliveto, Andy Zaidman, and Alexander Serebrenik. 2021. Beyond Technical Aspects: How Do Community Smells Influence the Intensity of Code Smells? *IEEE Transactions on Software Engineering* 47, 1 (2021), 108–129. https://doi.org/10.1109/TSE.2018.2883603

[48] Fabio Palomba and Damian Andrew Tamburri. 2021. Predicting the emergence of community smells using socio-technical metrics: a machine-learning approach. *Journal of Systems and Software* 171 (2021), 110847.

[49] Fabio Palomba and Damian Andrew Tamburri. 2021. Predicting the emergence of community smells using socio-technical metrics: a machine-learning approach. *Journal of Systems and Software* 171 (2021), 110847.

[50] Javier Portillo-Rodríguez, Aurora Vizcaino, Mario Piattini, and Sarah Beecham. 2014. Using agents to manage socio-technical congruence in a global software engineering project. *Information Sciences* 264 (2014), 230–259.

[51] Paul Ralph, Mike Chiasson, and Helen Kelley. 2016. Social Theory for Software Engineering Research. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering* (Limerick, Ireland) *(EASE '16)*. Association for Computing Machinery, New York, NY, USA, Article 44, 11 pages. https://doi.org/10.1145/2915970.2915998

[52] Ayushi Rastogi, Nachiappan Nagappan, Georgios Gousios, and André van der Hoek. 2018. 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*. 1–8.

[53] Ita Richardson, Valentine Casey, John Burton, and Fergal McCaffery. 2010. *Global Software Engineering: A Software Process Approach*. Springer Berlin Heidelberg, Berlin, Heidelberg, 35–56. https://doi.org/10.1007/978-3-642-10294-3_2

[54] Karlene H Roberts and Nakiye A Boyacigiller. 2012. 3. Cross-national Organizational Research: The Grasp of the Blind Men. In *Societal Culture and Management*. De Gruyter, 51–69.

[55] Camila Sarmento, Tiago Massoni, Alexander Serebrenik, Gemma Catolino, Damian Andrew Tamburri, and Fabio Palomba. 2022. Gender Diversity and Community Smells: a Double-Replication Study on Brazilian Software Teams. In *International Conference on Software Analysis, Evolution and Reengineering*. IEEE.

[56] Hina Shah, Nancy J. Nersessian, Mary Jean Harrold, and Wendy Newstetter. 2012. Studying the Influence of Culture in Global Software Engineering: Thinking in Terms of Cultural Models. In *Proceedings of the 4th International Conference on Intercultural Collaboration* (Bengaluru, India) *(ICIC '12)*. Association for Computing Machinery, New York, NY, USA, 77–86. https://doi.org/10.1145/2160881.2160894

[57] Arndt Sorge. 1983. Review of Culture's Consequences: International Differences in Work-Related Values. *Administrative Science Quarterly* 28, 4 (1983), 625–629. http://www.jstor.org/stable/2393017

[58] Viktoria Stray and Nils Brede Moe. 2020. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. *Journal of Systems and Software* 170 (2020), 110717. https://doi.org/10.1016/j.jss.2020.110717

[59] Yuri Takhteyev and Andrew Hilts. 2010. Investigating the geography of open source software through GitHub. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.463.1444.

[60] Damian Andrew Tamburri. 2019. Software Architecture Social Debt: Managing the Incommunicability Factor. *IEEE Transactions on Computational Social Systems* 6, 1 (2019), 20–37. https://doi.org/10.1109/TCSS.2018.2886433

[61] Damian Andrew Tamburri, Rick Kazman, and Hamed Fahimi. 2016. The architect's role in community shepherding. *IEEE Software* 33, 6 (2016), 70–79.

[62] Damian Andrew Tamburri, Philippe Kruchten, Patricia Lago, and Hans van Vliet. 2013. What is social debt in software engineering?. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 93–96.

[63] Damian Andrew Tamburri, Philippe Kruchten, Patricia Lago, and Hans van Vliet. 2015. Social Debt in Software Engineering: Insights from Industry. *Journal of Internet Services and Applications* (2015). https://doi.org/10.1186/s13174-015-0024-6

[64] Damian Andrew Tamburri, Fabio Palomba, and Rick Kazman. 2019. Exploring Community Smells in Open-Source: An Automated Approach. *IEEE Transactions on Software Engineering* 47, 3 (2019), 630–652. https://doi.org/10.1109/TSE.2019.2901490

[65] Damian Andrew Tamburri, Fabio Palomba, Alexander Serebrenik, and Andy Zaidman. 2019. Discovering community patterns in open-source: a systematic approach and its evaluation. *Empirical Software Engineering* 24, 3 (2019), 1369–1417.

[66] Giuseppe Valetto, Mary Helander, Kate Ehrlich, Sunita Chulani, Mark Wegman, and Clay Williams. 2007. Using software repositories to investigate socio-technical congruence in development projects. In *Fourth International Workshop on Mining Software Repositories (MSR'07: ICSE Workshops 2007)*. IEEE, 25–25.

[67] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. 2015. Perceptions of Diversity on Git Hub: A User Survey. In *8th IEEE/ACM International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015, Florence, Italy, May 18, 2015*, Andrew Begel, Rafael Prikladnicki, Yvonne Dittrich, Cleidson R. B. de Souza, Anita Sarma, and Sandeep Athavale (Eds.). IEEE Computer Society, 50–56. https://doi.org/10.1109/CHASE.2015.14

[68] Bogdan Vasilescu, Alexander Serebrenik, and Vladimir Filkov. 2015. A Data Set for Social Diversity Studies of GitHub Teams. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. 514–517. https://doi.org/10.1109/MSR.2015.77

[69] Ramya T Venkateswaran and Abhoy K Ojha. 2019. Abandon Hofstede-based research? Not yet! A perspective from the philosophy of the social sciences. *Asia Pacific Business Review* 25, 3 (2019), 413–434.

[70] Laurie Williams and Robert R Kessler. 2003. *Pair programming illuminated*. Addison-Wesley Professional.

[71] Claes Wohlin, Martin Höst, and Kennet Henningsson. 2003. Empirical research methods in software engineering. In *Empirical methods and studies in software engineering*. Springer, 7–23.