An Empirical Investigation into the Influence of Software Communities' Cultural and Geographical Dispersion on Productivity

Stefano Lambiase,[1] Gemma Catolino,[2] Fabiano Pecorelli,[2] Damian A. Tamburri,[2] Fabio Palomba,[1]
Willem-Jan van den Heuvel,[2] Filomena Ferrucci[1]

[1]*SeSa Lab - University of Salerno (Italy)* – [2]*Jheronimus Academy of Data Science (The Netherlands)*

Abstract

Estimating and understanding software development productivity represent crucial tasks for researchers and practitioners. Although different works focused on evaluating the impact of human factors on productivity, a few explored the influence of cultural/geographical diversity in software development communities. More particularly, all previous treatise addresses cultural aspects as abstract concepts without providing a quantitative representation. Improved knowledge of these matters might help project managers to assemble more productive teams and tool vendors to design software analytics toolkits that may better estimate productivity. This paper has the goal of enlarging the existing body of knowledge on the factors affecting productivity by focusing on *cultural and geographical dispersion* of a development community—namely, how diverse a community is in terms of cultural attitudes and geographical collocation of the members who belong to it. To reach this goal, we performed a mixed-method empirical study. First, we built a statistical model relating dispersion metrics with the productivity of 25 open-source communities on Github. Then, we performed a confirmatory survey with 140 practitioners. The key results of our study indicate that cultural and geographical dispersion considerably impact productivity, thus encouraging managers and practitioners to consider such aspects during all the phases of the software development lifecycle. We conclude our paper by elaborating on the main insights from our analyses and instilling implications that may drive further research.

Keywords: Global Software Engineering, Social Aspects in Software Engineering, Productivity, Empirical Software Engineering.

1. Introduction

Software productivity refers to the efficiency and effectiveness of software development teams in creating, testing, and deploying software systems [1]. High productivity levels typically lead to faster time-to-market, better quality software, and greater customer satisfaction [2, 3]. Nonetheless, assessing software productivity is challenging: while different productivity indicators have been proposed, *e.g.*, lines of code written or bugs fixed per time unit [4–6], the perception and feeling of productivity can vary among team members [7], *e.g.*, a developer may feel highly productive when accomplishing difficult tasks rather than based on the time spent on coding.

Global Software Engineering (GSE) [8, 9]—*i.e.*, the set of practices and guidelines aimed at managing distributed teams—further challenges the estimation of development productivity. Indeed, in such a distributed scenario additional social and cultural aspects come into play [10, 11]. For instance, a sub-optimal management of the heterogeneity of development teams may cause heavier inter-team communication and misunderstanding among team members [9, 12, 13] that altogether negatively affect software communities' productivity [14–19].

The GSE research community has been conducting multiple studies targeting the relation between social and human factors on productivity [20–23], other than the underlying reasons behind miscommunication, *e.g.*, community smells [24, 25] and gender diversity [26, 27]. While these previous works have provided compelling evidence of how productivity can be affected by social and human aspects, our research highlights a noticeable lack of knowledge of the role played by *cultural and geographical dispersion*, which are factors that can significantly impact how individuals perceive and measure productivity. Some cultures indeed emphasize teamwork, while others value individual effort [28, 29]: these differences may lead to variations in how team members perceive productivity and their overall effectiveness. Furthermore, geographically dispersed teams may experience concerns due to communication barriers and time zone differences, which might lead to inaccurate performance evaluations and hinder team effectiveness.

An improved understanding of the impact of cultural and geographical dispersion on productivity may lead to significant improvements in multiple project management knowledge areas and procedures [30]. For example, knowing more about the dispersion of a development team since the initiation processes could lead to better risk management planning and a consequent more robust risk register, mitigation, and contingency strategies. Consequentially, better risk management knowledge positively impacts all the other knowledge areas—e.g., cost, time, and communication—leading to a general improvement of the overall software development process.

In our previous empirical study [31], we took a first step toward this direction by conducting a quantitative investigation on the relation between the cultural and geographical dispersion of a community—*i.e.*, the degree to which a community is formed by individuals growing up in and coming from different places globally—and its productivity, as measured by the number of commits in a specific time range [31]. We represented culture using the *Hofstede 6D Model*, namely, a framework that uses six measures to uniquely represent the culture of individuals based on their origin country, widely known and adopted for various work in the software engineering field [32–35]. We built a statistical regression model to assess the relationship between the two dispersion metrics and productivity, conducting the study on 25 open-source communities on Github. The results of our analysis showed that dispersion metrics influence developers' productivity both positively and negatively.

In this paper, we extend our previous work [31] by complementing the statistical study with a *qualitative* empirical examination of the perceived value and impact of dispersion metrics over the productivity of software communities, hence assessing the overall goals of our study through *mixed-method research* [36]. More particularly, we complemented the statistical modeling findings with a survey study involving 140 participants with experience in distributed software development. By means of the survey study, we could first corroborate the statistical investigation with more qualitative insights aiming at understanding the value of cultural and geographical dispersion for the overall productivity of software communities and strengthen our findings. Secondly, we could also elaborate and further analyze a potential threat to the validity of the previous study, *i.e.*, we estimated productivity through the number of commits within a time range; however, developers might perceive productivity differently, hence influencing the conclusions drawn in our preliminary study.

In the first place, our findings confirm the results achieved in our preliminary investigation [31]: cultural and geographical dispersions sensibly impact a software development community's productivity, but not necessarily in a negative manner. Depending on the specific dispersion metrics considered in our study, we could draw implications for productivity. Our survey participants also confirmed the soundness of the design decisions taken in the statistical modeling exercise: the number of commits per time range is perceived as a valid proxy to estimate the actual community productivity. As a further result, both the survey and the statistical model confirm that socio-technical factors—*e.g.*, team size and truck factor—influence the productivity of a software development community.

To sum up, our work provides the following contributions:

1. Statistical insights into the role of dispersion metrics on software productivity;
2. A large-scale confirmatory survey study involving 140 developers to qualitatively validate the results of the statistical model and strengthen them;
3. An online replication package [37] publicly available to support replication and future work.

Our work finally offers and discusses a number of insights and implications to researchers, practitioners, and tool vendors, on how to effectively exploit cultural and geographical metrics to improve the overall productivity of software teams.

Structure of the paper. Section 2 describes the background of this work and the related work. In Section 3, we present and outline the design of our study, and Section 4 reports the study results. Section 5 discusses the insights of the paper, and Section 6 examines the threats to the validity and how we mitigated them. Finally, Section 7 concludes the paper and provides insights on our future research agenda.

2. Background and Related Work

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

2.1. Cultural Aspects in Software Engineering

Nowadays, software development is often a geographically distributed effort involving stakeholders and practitioners collaborating from different places worldwide [10, 11, 38]. For such a reason, *Global Software Engineering* (GSE)—*i.e.*, the application of software engineering practices for managing and developing software distributed projects [8, 38–40]—and the associated research field are becoming even more popular. Specifically, in the large set of topics the research community discusses, the social impact of such *"dispersion"* on product and project metrics is of particular interest. In particular, among the various problems, there is a growing interest in characterizing cultural aspects and their potentially catastrophic impact if not correctly managed [38, 40–42].

Culture has been defined by Kreitner et al. as a set of granted assumptions about how to act and think that characterize a community of individuals [43]. Indeed, culture is a complex topic and is difficult to formalize and measure, so most of the work in literature treats it in an abstract manner [41]. Nevertheless, having a tool to measure culture quantitatively is beneficial to conduct research [44]. For such a purpose, various frameworks and tools arise to allow researchers and practitioners in various fields to measure and represent the culture of individuals [29, 45, 46]. Most of them consisted of a representation based on a set of numerical values—called *dimensions*—whose combination uniquely characterizes the cultural behavior of an individual or a group of people. For example, Hampden-Turner et al. [46] represented culture using three layers, *i.e.*, *explicit culture*, *norms and values*, and *assumption about existence*.

Of particular interest to our work is the framework proposed by Geert Hofstede, *i.e.*, the *Hofstede 6D Framework*, a set of six dimensions that assume values from zero to one hundred and which combination characterizes individuals from a specific country globally[8, 39, 41, 42]. The six Hofstede's dimensions are defined as follows:

Power Distance Index (*PDI*). It refers to the degree of inequality that exists and is accepted between people with and without power in a community. A high score of *PDI* indicates that society accepts a hierarchical order in which everybody has a determined place. On the contrary, in societies with low

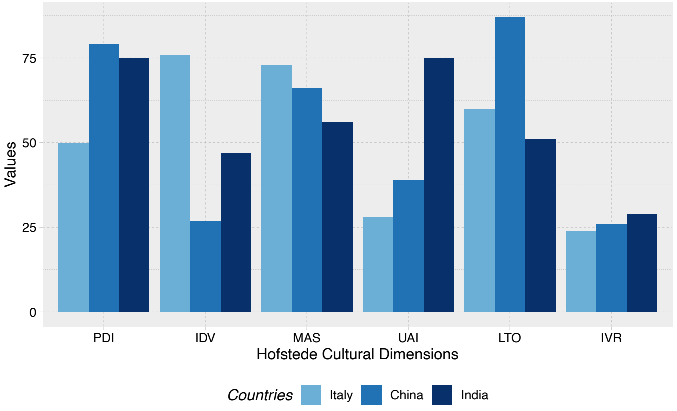

Figure 1: Hofstede's cultural dimensions values computed for Italy, China, and India.

PDI, people seek to equalize the power distribution between all community members.

Individualism vs. Collectivism (*IDV*). It represents the degree to which people in a society are integrated into groups and their perceived obligations and dependence on groups. A high level indicates a society where individuals are expected to care for only themselves and their immediate families. Conversely, a low level indicates a society in which people are supposed to be loyal to their group.

Masculinity vs. Femininity (*MAS*). It represents a contrast between the two preferences. The Masculinity side indicates a preference for achievement, heroism, assertiveness, and material rewards for success. In contrast, the Femininity side represents a preference for cooperation, caring for the weak, and quality of life.

Uncertainty Avoidance (*UAI*). It expresses the degree to which the members of a society feel uncomfortable with uncertainty and ambiguity. A high level of *UAI* indicates that people tend to (1) maintain rigid codes of belief and behavior and (2) 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.

Long vs. Short Term Orientation (*LTO*). It measures how much people are oriented toward a long-term vision of life than a short-term one. A high score of *LTO* indicates that people emphasize persistence, perseverance, and long-term growth. On the contrary, a low score of *LTO* indicates that people emphasize quick results and respect for tradition.

Indulgence vs. Restraint (*IVR*). It refers to the extent and tendency of a society to fulfill its members' desires. A high level indicates that society allows relatively free gratification related to enjoying life and having fun. Conversely, a low level indicates a society that controls the gratification of needs and regulates it using strict social norms.

To better contextualize the Hofstede framework, Figure 1 shows an example of the dimensions computed for three countries, *i.e.*, China, Italy, India.[1] For example, looking at the IDV score value of Italy, we can notice that the score is 75, representing a society in which individualism tend to be preferred to collaboration. On the contrary, China has a score of 28, meaning people act in the interests of the group and not necessarily of themselves. Moreover, analyzing the UAI scores, we can see that Italy and China have a value of 30 and 40, indicating that people in such countries tend to be more tolerant of risky situations and changes. In contrast, India has a value of 75, revealing a more traditional society.

Despite the diffuseness of this framework, the research community had mixed opinions on its validity [47–49] supporting in some cases its rejection [50–52]. For instance, Brewer and Venaik [50, 51] deemed unreliable the tool's ability to represent cultural profiles. Nonetheless, follow-up studies highlighted the potential of the framework; for example, Venkateswaran and Ojha [35] showed that Hofstede's framework represents the most effective way to characterize the complex world of cultural aspects which has been already shown effective in several fields [32–34], *e.g.*, management, law, politics, ethics, architecture, medicine, and computer science [53].

Some works studied the use of Hofstede in the context of software engineering [32, 54]. Recently, Darwish and Henryson [54] published a thesis that studied how cultural background influences adopting a specific SE practice (e.g., documentation design, refactoring, test driven development). They reported that developers with similar behaviors related to cultural dimensions (i.e., PDI and UAI) tend to adopt similar practices (e.g., making early design decisions and test driven development). In another work, Borchers [32] conducted research on how cultural factors influence software engineering processes, such as code review. The study specifically examined three countries: Japan, India, and the United States. The findings revealed that various cultures approach software engineering processes differently. For instance, Japanese developers exhibit a significant level of UAI, which leads to slower decision-making. Borchers also emphasized that cultural differences within software teams can affect software architecture, suggesting further exploration of this topic in the field.

2.2. *Productivity Factors*

Productivity is a complex concept to define and measure. Nevertheless, various metrics arise [17, 56–59], and researchers agree that productivity measures should be expressed in terms of output produced in a given time given a specific input [57, 59]. For example, some works define the productivity of a development community as the number of accomplished contributions by team members—*e.g.*, commits, push, or tasks completed in a given unit of time—for the entire project duration [59–61].

[1] Hofstede 6D Model website: `https://www.hofstede-insights.com/country-comparison/china,india,italy/`

Table 1: Comparison with Related Work.

Related Work	Main Focus	Differences
Wagner and Ruhe [15]	It is a systematic literature review showing how product metrics such as code reuse, software size, and programming language highly impact developers' productivity.	• No focus on social metrics has been put in the investigation. • No quantitative or qualitative methods have been used to validate the findings.
Graziotin et al. [18]	They performed an observational study with eight participants to study the influence of affecting dimensions on self-assessed productivity.	• Paper did not investigate cultural and dispersions metrics. • Authors did not use quantitative methods.
Vasilescu et al. [19]	They performed a mixed-method approach study to investigate the influence of gender and tenure diversity on team productivity.	• Paper did not investigate cultural and dispersions metrics. • Authors did not focus on productivity measures.
Murphy-Hill et al. [55]	They surveyed practitioners from three private companies to identify factors that can influence self-assessed productivity.	• Paper did not investigate cultural and dispersions metrics. • Authors did not focus on productivity measures. • Authors did not use quantitative methods.
Darwish and Henryson [54]	They studied how cultural background influences adopting a specific SE practice aiming to provide insights that allow managers to take advantage of each culture's strengths.	• The context of the study was limited to Indonesian and Sweden developers. • The authors did not study the impact of culture on productivity.
Borchers [32]	It is a report and analysis of the author's past experience in managing software teams composed of people from Japan, India, and the United States. The goal was to study how different cultures approach software development phases.	• The author used only three of the six Hofstede Dimensions to characterize culture. • The context of the study was limited to Japan, India, and the United States developers. • The authors did not study the impact of culture on productivity.

Additionally, researchers investigated which factors influenced the productivity of a software development team [14–19, 62]. From our review, we can divide these studies based on the metrics found relevant, *i.e.*, technical and social:

Technical Factors. Regarding technical factors, *i.e.*, product metrics and tools, Wagner and Ruhe [15] conducted a systematic review, showing how metrics like code reuse, software size, and programming language highly impact developers' productivity. Finally, Mohagheghi and Conradi [16] investigated the relationship between productivity and software reuse, showing positive results.

Social Factors. Regarding social factors—mainly related to people and their relations—Murphy-Hill et al. [14] surveyed practitioners, showing how social factors—*e.g.*, people's enthusiasm, peer support, and valuable feedback about job performance—strongly affect people's productivity. Moreover, Wagner and Ruhe [15] demonstrated that social factors like corporate culture and working environment are essential to enhance software teams' productivity. Graziotin et al. [18] showed how valence and dominance dimensions in developers affect self-assessed productivity. Finally, Vasilescu et al. [19] demonstrated how gender and tenure diversity are good predictors of productivity using a statistical model. Moreover, they adopted a mixed-method approach—*i.e.*, data mining and surveys—to strengthen their findings.

More recently, Murphy-Hill et al. [55] studied the concept of *self-rated productivity* [63] and the factors able to influence it. Specifically, they conducted a survey with practitioners from private companies in IT fields—*i.e.*, Google, ABB, and National Instruments—to identify and study the aspects that practitioners could use to predict the productivity of a software development team. Their findings suggest that the factors most useful to predict productivity are not technical but social ones—*e.g.*, job enthusiasm, peer support for new ideas, and receiving helpful feedback about job performance.

With respect to the works discussed so far, we identified multiple differences and research gaps. Table 1 summarizes the major characteristics of the related work and points out the main research gaps that our study aims at filling.

Graziotin et al. [18] and Vasilescu et al. [19] conducted investigations on social and human factors on productivity, not considering cultural and geographical indicators. Our study can be therefore seen as complementary, as it enlarges the body of knowledge with respect to these previous papers.

As for Murphy-Hill et al. [14] and Wagner and Ruhe [15], they conducted more general survey studies which did not have the explicit goal of analyzing the impact of culture. In other terms, the findings reported on the matter can be considered tangential and not comprehensive. Moreover, our mixed-method investigation allowed us to investigate cultural and geographical dispersion under different perspectives, hence strengthening the conclusion validity and generalizability of the findings.

Perhaps more importantly, the authors treat cultural aspects in an abstract and high-level manner without operationalizing them. On the contrary, we exploited Hofstede's 6-D framework [28], which was explicitly defined to represent cultural dimensions quantitatively.

Last but not least, Murphy-Hill et al. [55] assessed *self-rated productivity* [63], while we experimented with the number of commits per time range. This metric was considered a valid proxy for productivity by the practitioners involved, hence potentially representing an additional insight provided by our work with respect to how productivity can be estimated.

3. Research Study Design

This section describes the research questions and the methods used to achieve the study's primary objective.

3.1. Research Questions and Goals

The *goal* of the study is to analyze the relationship between the *cultural and geographical dispersion* of a development community and its productivity, computed by counting the number of commits in a range of time [57]. The *purpose* is to increase awareness and allow practitioners to make more informed decisions based on their software development community. The *perspective* is of managers interested in effectively allocating resources, adhering to the project's requirements, or managing/monitoring complex organizational structures.

The goals of the empirical study were mapped onto the following research question.

> **RQ – Dispersion Metrics Versus Productivity**
>
> *To what extent do cultural and geographical dispersion influence teams' productivity?*

Figure 2 overviews the methods employed to address our research question. We adopted a mixed-method research approach [36] in which both qualitative and quantitative studies are performed to reach theoretical saturation. Specifically, in our research, we performed the following studies:

Quantitative Investigation: we built a statistical model—a mixed-linear regression model [64]—able to assess whether cultural and geographical dispersion relates to the productivity of open-source development communities.

Qualitative Investigation: we surveyed 140 practitioners with experience in distributed software development to gather their opinion on how different culturally originated behaviors in software teams can influence their productivity.

In terms of reporting, we employed the guidelines by Wohlin et al. [65], other than following the *ACM/SIGSOFT Empirical Standards*.[2] Moreover, a replication package of the study is available online [37].

Table 2: Projects in the dataset.

Project	Progr. Language	# Windows
Akretion	Python	6
Bigcheese	C++	1
Burke	Go	5
Chapuni	C++	9
Cloudfoundry	Shell	7
CTSRD-CHERI	C++	2
Django	Python	23
Emberjs	Python	7
Fangism	C++	1
Genome	Perl	5
Holman	C	7
Jedi4ever	Shell	8
Jrk	C++	1
Liferay	Java	12
Loganchien	C++	1
Moodle	PHP	14
Mozilla - gecko-dev	C++	1
Mozilla - OpenBadger	Javascript	2
Mxcube	Python	2
Puppetlabs	Ruby	14
RobbyRussel	Python	15
Rspec	Ruby	13
Symfony	Python	13
Torvalds	C	17
Travis-ci	Javascript	10

3.2. Design of the Quantitative Study: Regression Model

The first step of our study consisted of a statistical analysis analyzing the relationship between dispersion metrics—*i.e.*, cultural and geographical dispersion—and the productivity of software development communities. In the following, we provided information about our quantitative investigation.

3.2.1. Data Collection

To conduct our study, we used the dataset from our previous study [66] containing socio-technical metrics about 25 open-source software communities. Specifically, the dataset contains information for different time windows, made of 90 days. Therefore, we had information in various time slices for each software development community. Table 2 reports the list of the projects and the number of windows considered for each of them in our study.

Within the dataset, the most valuable metrics are represented by software communities' cultural and geographical dispersion indicators. As for cultural dispersion, the dataset contains six cultural metrics that can assume values from zero to fifty. Each

[2] Available at the following link: `https://github.com/acmsigsoft/EmpiricalStandards`. Given the nature of our study and the currently available standards, we followed the *"General Standard"*, *"Questionnaire Surveys"*, *"Mixed Methods"*, and *"Data Science"* definitions and guidelines.

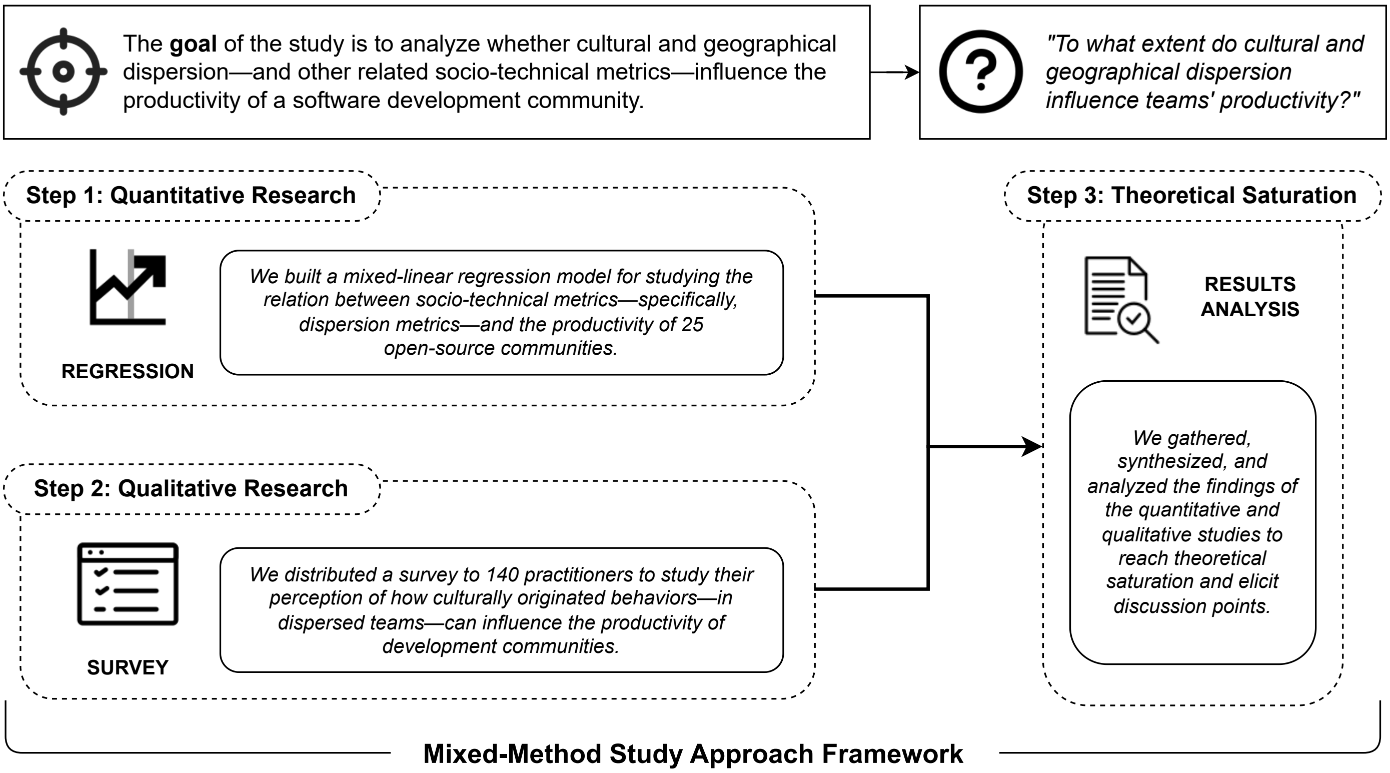
The goal of the study is to analyze whether cultural and geographical dispersion—and other related socio-technical metrics—influence the productivity of a software development community.
"To what extent do cultural and geographical dispersion influence teams' productivity?"
Step 1: Quantitative Research
REGRESSION
We built a mixed-linear regression model for studying the relation between socio-technical metrics—specifically, dispersion metrics—and the productivity of 25 open-source communities.
Step 2: Qualitative Research
SURVEY
We distributed a survey to 140 practitioners to study their perception of how culturally originated behaviors—in dispersed teams—can influence the productivity of development communities.
Step 3: Theoretical Saturation
RESULTS ANALYSIS
We gathered, synthesized, and analyzed the findings of the quantitative and qualitative studies to reach theoretical saturation and elicit discussion points.
Mixed-Method Study Approach Framework

Figure 2: Overview of the research methods used to address the research questions of the study.

metric corresponds to the standard deviation of the set containing the community members' value for one of the six dimensions of Hofstede [28]. As for the geographical dispersion, the metric considered the standard deviation of the spherical distances (in miles) between each community member as a metric—computed using the GeoPy[3] library. To compute both metrics, we relied on the original country of the developers in the development communities, which was provided in the original dataset [26, 67]—more details in the following section.

3.2.2. Statistical Model Variables

To answer our research question, we built a statistical *linear mixed regression model* [64] relating a development community's cultural and geographical dispersion to productivity—expressed in terms of the number of commits.

Independent Variables. In the context of our study, we considered the following independent variables.

Cultural Dispersion. The *cultural dispersion* of a development community indicates how such a community is formed by developers coming from different cultural behaviors [66, 68]. Being a quantitative measure, it is necessary to represent the culture of individuals using a quantitative framework; hence, we used the six metrics in Hofstede's framework [28]—described in Section 2. By such a choice, we defined six cultural dispersion metrics—one for each Hofstede dimension—corresponding to the standard deviation of the set containing the Hofstede community members' values. The metrics are reported in the following:

- **PDID**: *Power Distance Index Dispersion* indicates how much community members tend to have a different idea of how power should be distributed between them.
- **IDVD**: *Individualism vs. Collectivism Dispersion* indicates how much community members tend to have different ideas regarding working in a group and sharing success or being individualistic.
- **MASD**: *Masculinity vs. Femininity Dispersion* indicates how much community members tend to have a different opinion about self-affirmation and help the weaker elements.
- **UAID**: *Uncertainty Avoidance Dispersion* indicates how much community members tend to have different ideas on taking risks and accepting new and controversial opinions.
- **LTOD**: *Long Term Orientation vs. Short Term Orientation Dispersion* indicates how much community members tend to have a different opinion about investing or not in the future and conserving old traditions and habits.
- **IVRD**: *Indulgence vs. Restraint Dispersion* indicates how much community members tend to have a different opinion about the rank in which the governing authority controls how people satisfy their needs and spend leisure.

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

Geographical Dispersion. The *geographical dispersion* of a development community indicates how such a community is formed by developers collaborating from different places geographically distributed around the globe. As shown in Equation 1, we operationalized the geographical dispersion of a community as the standard deviation of the set containing the physical distances—expressed using spherical distance—between each community member [68, 69].

$$GeographicalDispersion(X) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu_X)^2} \quad (1)$$

where X is the set containing the physical distance between each pair of community members, N is the number of elements in X, X_i is the element of index i in X, and μ_X is the mean of X.

To provide an example for better explaining this metric, given a development community: (1) if all the members are working from different and distant locations, there is a high level of geographical dispersion; (2) if the members are working in clusters geographically distributed, there is a medium level of geographical dispersion; (3) if the members are all working in the same office, there is a zero level of geographical dispersion.

Geographical Dispersion Example

To provide an example of the adoption of the formula, let's consider the case in which we have a community of 15 practitioners. If they are separated into 3 clusters of the same size—i.e., Amsterdam, India, and Seattle—we have a set of 105 distances (the pairs of 15 items without repetition). Of these, 30 are 0 km value, 25 are 7849 km (distance between Seattle and Amsterdam), 25 are 12158 km (distance between India and Seattle), and 25 are 7146 km (distance between India and Amsterdam). The geographical dispersion is 4772,24.

We computed these metrics for each time window presented in our dataset. We make this choice because community members can change over time.

Dependent Variable. Since our goal was to understand the impact of cultural and geographical dispersion on the productivity of a development community, we used the number of commits per time [59–61] as a reference measurement. We relied on this metric since it has been widely used when dealing with productivity in past works [59–61, 70, 71].

Control Variables. When constructing a statistical model, it is essential to consider that beyond independent variables, several variables can affect the phenomenon analyzed, as demonstrated in the literature [18, 19, 26, 66, 72, 73]. For this reason, we considered the following variables:

- **Number of Committers**: It is defined as the number of people that have done at least one commit in a given project time window. Having more committers could imply high productivity in terms of the number of commits.

- **Team Size**: It represents the number of contributors per team in a given temporal window. The community's size can influence the number of commits done during the development of the project.

- **Turnover**: It concerns the fraction of the team in a given temporal slice that is different from the previous windows (*i.e.*, the *turnover ratio*). A high turnover means that team members change frequently. The constant introduction of new members might lead to the variability of productivity.

- **Project Age**: It represents the difference between the maximum index and the index of the 90-day temporal interval from the first commit. Older projects and their teams could have low productivity since their systems are running into a maintenance phase and not a developing phase that is generally more active.

- **Tenure diversity**: *Tenure measure* is defined as the experience of developers in various fields [74], thus possibly affecting productivity [19]. In our study, we considered two types of tenure: (1) *commit tenure* (that represents the coding experience of a contributor within all GitHub projects in which s/he contributed), and (2) *project tenure* (that represents the contributor's experience in the specific project considered).

- **Tenure median**: It represents the median project tenure and the commit median tenure and is used to complement *tenure diversity*.

- **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.

- **Blau-Index**: Blau [75] 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 [76] represents "the state in which a software development organization harbors sufficient coordination capabilities to meet the coordination demands of the technical products under development.".

- **Truck Factor**: TF represents the minimum number of members of a team that have to quit before the project fails [77–80].

- **Centrality**: It is defined as the strength of a community, and it is based on modularity measures [81]. A value over 0.3 means that the community is highly modular, thus clearly distinguishing the sub-communities in its development network. A value below 0.3 means that there are no sub-communities instead.

The above factors were considered as control variables in our statistical models, according to previous literature [18, 19, 26, 66, 72, 73].

3.2.3. Statistical Model Construction

The dataset used [66] consists of multiple temporal windows for each project analyzed. This means there are multiple snapshots of the community's situation for the same team, i.e., our data expose a hierarchical structure (based on the team), then non-independence. For this reason, we constructed a *linear mixed model* able to capture measurements from within the same group (*i.e.*, within the same team) as a random effect [64].

Linear mixed models [64] extend simple linear models by incorporating both fixed and random effects. They are particularly employed when dealing with nonindependence in the data, often arising from hierarchical structures. Mixed models comprise fixed effects, which are parameters that remain constant, and random effects, which are parameters that are treated as random variables. Such a model has started to be adopted in software engineering research when analyzing the influence between variables in cases multiple snapshots for the same study item are provided [26, 82, 83]. In particular, we used the time window as a random effect and the rest of the above variables (in Section 3.2.2) as fixed effects. From an implementation perspective, we relied on the functions `lmer` and `lmer.test` available in the `R` package `lme4` [84].

It is important to note that we also faced the problem of multi-collinearity [85], which happens when an independent variable is highly correlated with one or more of the other independent variables, thus affecting the reliability of the results. For this reason, we used a stepwise variable removal process based on the *Companion Applied Regression* (`car`) `R` package[4], using the `vif` function [85].

To support our results, we computed the effect sizes of the coefficients using the well-known ANOVA statistical test [86]. Variables are considered significant if they are statistically significant, *i.e.*, the *p*-value is less than 0.05. Finally, for the sake of results reliability, we built two baseline statistical models—the first one containing all the control variables and the random effect—comparing them through the *AIC* (*Akaike information criterion*) and *BIC* (*Bayesian information criterion*) [87, 88] estimators. They are used for assessing the quality of the prediction [89]. Indeed, both estimate the prediction error and quality of statistical models for a given set of data. Models with a low value of AIC and BIC is the one that better characterizes the sample analyzed. This comparison allows us to study whether adding the independent factors improves the model's capability to estimate software development productivity; the comparison with the second model is whether the obtained results reflected the random effect instead.

3.3. Design of the Qualitative Study: Survey Design

The second step of our study consisted of a qualitative investigation to study how culturally originated behaviors, and geographical dispersion can influence the productivity of a software development community. Specifically, we surveyed 140 practitioners with experience in distributed software development and management. We adopted a survey as a qualitative method to study productivity based on previous evidence that showed how (i) they are straightforward and commonly used methods to measure self-assessed productivity [14, 90] and (ii) they are flexible methods to collect general insight from industry [91]. The following sections provide information about our qualitative research approach and dissemination actions.

3.3.1. Survey Structure

Our survey consisted of eight main sections described in the following paragraphs.

Survey Initiation Section. The first section was inspired by the work of Murphy-Hill et al. [14] and aimed at providing a baseline for all respondents regarding the definition of productivity—to obtain more cohesive results. Moreover, we extracted insights from participants on how to measure software development team productivity.

Survey Core Section. The sections from the second to the seventh were mapped on the six Hofstede dimensions and aimed at extrapolating the perceived impact of cultural dispersion metrics on self-assessed productivity. We developed such sections—and associated questions—using a *vignette-based scenario approach* [92], mainly used in psychological and sociological experiments [93]. Specifically, each scenario described the characteristics of one of the cultural dimensions contextualized in the software development environment.

For example, the box in the following shows the vignette used to describe the characteristics of the *Power Distance Index (PDI)* dimension:

Example Vignette Scenario

"Suppose your development team is working on the definition of an E-commerce application. During the development, you recognize the presence of people who (1) demand to equalize the distribution of the power between all team members, in contrast to others who (2) want to follow rigidly hierarchical organization."

In the above example and all the scenarios, we described two opposite behaviors, represented using the (1) and (2) items. We operationalized such a choice to allow participants to visualize better the situation we were interested in investigating [93]. All the used scenarios are available in our online appendix [37].

In order to better explain this choice, it is essential to stress that each of Hofstede's dimensions describes two opposite behaviors. For example, regarding *Individualism vs. Collectivism*, a high level indicates a society where individuals are expected to take care of only themselves and their immediate families.

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

In contrast, a low level indicates a society in which people are supposed to be loyal to their group. In the context of dispersion metrics, we are interested in how people characterized by two opposite behaviors coexist in the same community. For such a reason, we decided to use the strategy of the scenario described above. Moreover, to avoid bias in participants' answers, we described the two behaviors directly instead of referring to the culture of individuals—for this reason, in the following section, we introduce the concept of 'culturally originated behaviors'.

After each vignette, we asked questions aimed at understanding (1) whether the situation has ever happened to our participants in their past projects (Likert Scale from *Never* to *Always*); (2) to what extent the cultural background and physical distance influence the emergence of such behaviors (Likert Scale from *Not at all* to *To a Great Extent*); (3) how much the contrast between the behaviors impact productivity (Multiple choice grid with behaviors and Likert Scale from *Strongly Disagree* to *Strongly Agree*).

We decided to use Likert Scale with five values because they are largely suggested by Kitchenham and Pfleeger [91] for qualitative studies in software engineering. Such a scale allows us to increase the readability of our survey and be exhaustive without the risk of misconception by participants. Moreover, choosing five as the number of values allowed us to extract results without being too dispersed.

Although only the third question directly relates to our research question and objective, we included the other questions mainly for three reasons. First, to be sure that the participants were getting more into the context described by the scenario; second, to have additional data to assess the goodness and quality of the answers given; and third, to gather valuable cross-sectional information in the discussion phase of the results.

Control Variables Validation. The eighth section of the survey contained a single question asking participants their opinion on the influence of some socio-technical factors on the productivity of a software development community. Specifically, we aimed to evaluate practitioners' perceptions of the control variables used to build our statistical model. For such a purpose, we asked participants to express if a particular factor could impact productivity, using a Likert scale from *Definitely not* to *Definitely*. The control variables included were:

- Team Size: the dimension of the team in terms of people.
- Turnover: the changing of the workforce assigned to a production process.
- Project Age: the number of years from the start of the project—*i.e.*, the first commit.
- Tenure diversity: the difference of experience between the various team members.
- Blau-Index: the team's diversity in terms of biological gender—*i.e.*, male and female.

Survey Demographic Section. The last section of the survey was reserved for demographic information. Our target population was composed of practitioners. We included questions related to job positions (i.e, Project Manager, Product Owner, Software Architect, Software Engineer, and other), gender, programming/management experience, and size of the team they were considering when answering the survey questions. Furthermore, we also asked participants about their cultural backgrounds. Once we had developed the survey, and before releasing it, we sought and obtained approval from the Ethical Board Committee of the University of the second author.

Survey Attention Question Section. Following the guidelines provided by Meade and Craig [94], we included in our survey an *attention check question* [94]—*i.e.*, a question aimed at checking if the participant is reading the questions and is not answering randomly. Specifically, the attention question is the following "Respond with 'Never' to this question.". We discarded three participants' answers that did not correctly answer them.

3.3.2. Survey Submission and Participants Characteristics

From a recruitment perspective, we carefully paid attention to the target population. Indeed, we used Prolific[5] to recruit experts. Prolific is a web-based platform to support researchers in finding participants for survey studies. The platform allows tuning the preference of surveyed, putting constraints, *i.e.*, people must be practitioners and experienced in distributed work. Prolific use an *opt-in* strategy [95]: this implies that participants get voluntarily involved, possibly leading to self-selection or voluntary response bias [96, 97]. We introduced an incentive of 3 dollars per valid respondent to mitigate this bias. Prolific automatically suggests this amount according to (i) the profiles selected and (ii) the survey time. Nonetheless, answers must be double-checked at the end of the survey before assigning the reward. The survey was made available from January 30 to February 22, 2023, and we surveyed 140 practitioners.

3.3.3. Survey Design Guidelines

During the design of the survey and its submission, we relied on various guidelines to improve our method and, consequentially, our findings. First, regarding the survey design, we relied on the guidelines provided by Kitchenham and Pfleeger [91] and Andrews et al. [98]—broadly adopted for software engineering qualitative studies. Additionally, we considered the scale and guidelines provided by Kitchenham and Pfleeger [91], thus allowing us to increase the readability of our survey. Finally, we took inspiration from previous qualitative investigations that focused on the productivity of software development teams [19, 55, 90].

Regarding the tool used for survey dissemination, *i.e.*, Prolific, we take inspiration from other work in literature [99, 100]. Specifically, we adopted insights provided by Reid et al. [99], which defined a series of recommendations to conduct surveys in the software engineering field using the platform.

[5] Prolific website: `https://www.prolific.co/`.

Following the guidelines by Flanigan et al. [101], we consciously kept the survey anonymous, preventing our influence on the answer. We created the survey as a Google form[6] and estimated a completion time of 10/15 minutes.

Regarding the analysis of the results, we used descriptive statistics/plots for the closed-ended questions of the surveys, while *content analysis* [102]—*i.e.*, a research method where one or more inspectors go over the data of interest and attempt to deduct their meaning and/or the concepts they let emerge—for open-ended questions. The process was conducted by the first three authors of the paper, who jointly analyzed the individual responses to identify and label the main insights and comments left by participants.

4. Analysis of the Results

This section illustrates the results of our study. For the sake of comprehensibility, we decided to divide the section into three subsections: the first one reporting the previous results from the quantitative analysis [31]; the second one reporting our findings from the analysis of the survey responses; and the third one, consisting of a sum-up and synthesizing of the results from both the studies. Such a report strategy is coherent with the mixed-method approach used for this study [36]. Due to space limitations and readability, detailed and raw results are available in the online appendix [37].

4.1. Quantitative Study: Regression Model

This section shows the results achieved when assessing the relationship between cultural and geographical dispersion and the productivity of a development community.

Table 3 and Figure 3 report the details of the statistical models. The first model (see the column "All Variables") shows the results achieved considering both confounding factors and independent variables. In particular, the *number of committers* and the *centrality* seem to impact the productivity of a development community greatly. Moreover, the *project age* and the *number of females* in the team significantly impact the dependent variable too.

As for the cultural and geographical dispersion, we can notice that the most significant variables are *Individualism vs. Collectivism Dispersion* and *Long Term Orientation Dispersion*, followed by *Power Distance Index Dispersion*. *Indulgence vs Restraint Dispersion* and *Geographical Dispersion* seem significantly impact productivity, too. Therefore, we can claim that dispersion metrics influence the productivity of a software development community.

To provide a preliminary interpretation of our results—similar to our previous study [66]—cultural and geographical dispersion influence the productivity of a development community both positively and negatively. As proof of this, *Individualism vs. Collectivism Dispersion*, *Indulgence vs. Restraint Dispersion*, and *Geographical Dispersion* impact the dependent

Table 3: Statistical Model Results.

Factor	All Variables		Conf. Variables		Random
	Estimate	*Sig.*	*Estimate*	*Sig.*	*Estimate*
(Intercept)	2.663		1.957		9.565
Number of Committers	0.788	***	0.899	***	
Project Age	-0.058	***	-0.051	**	
Turnover	0.649		0.092		
Blau Gender	3.256	.	5.289	**	
Tenure Median	-0.018		0.021		
Tenure Diversity	-0.001		-0.001		
Team Size	0.301	.	0.401	*	
Socio-Technical Congruence	0.109		0.171		
Truck Factor	0.021		-0.003		
Number of Females	-0.055	*	-0.055	*	
Expertise	0.018		0.059		
Centrality	0.587	***	0.563	**	
PDID	-0.084	**			
IDVD	0.109	***			
MASD	-0.024				
UAID	0.0158				
LTOD	-0.111	***			
IVRD	0.088	*			
GeoD	0.001	*			

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

Table 4: AIC and BIC for the three models.

Metric	All Variables	Conf. Variables	Random
AIC	502	536	617
BIC	572	584	626

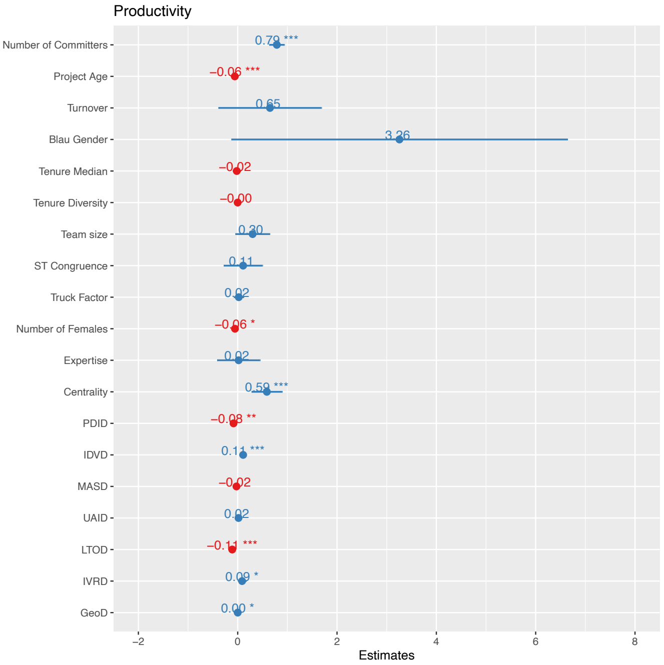

Figure 3: Statistical Model Representation.

[6] Google form website: https://www.google.com/forms/about/

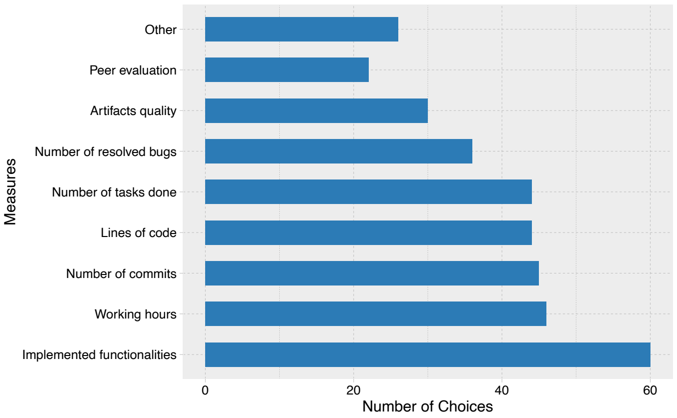

Figure 4: Productivity measures.

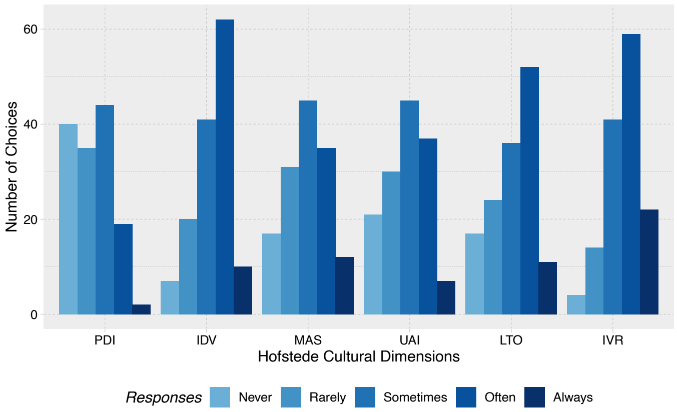

Figure 5: Frequence of experienced cultural behaviors.

variable positively, while *Power Distance Index Dispersion* and *Long Term Orientation Dispersion* negatively.

Table 4 shows the AIC and BIC value for the three models. The model with all variables had the lowest index value, i.e., 502 and 572, compared to the one with only control and random variables. Thus, adding the independent variables contributes to explaining productivity better.

Quantitative study: summary of the results.

Our study confirmed how socio-technical metrics could significantly impact the productivity of a development team. In addition, the study revealed that culture and geographical dispersion influence the productivity of a software development community.

4.2. *Qualitative Study: Survey*

The following section reports (1) the background and demographic information of the study's participants and (2) the findings of our qualitative analysis.

4.2.1. *Participants Background Information*

Among the 140 participants, 70% of them worked as developers, 15% as Project Managers, and the remaining had other roles, e.g., software architect and data scientist. Moreover, 50% of our sample was male, while 50% was female. Indeed, during the survey definition, we precisely decided to balance our sample based on gender in order to collect opinions from both of them. Furthermore, most of our participants self-assessed their management and programming skills as high or medium, 52% and 80%, respectively. From these basic descriptive statistics, we can claim that the answers collected provide reliable insights for validating the information gathered from the survey.

4.2.2. *Perception of Productivity Measurement by Developers*

For our quantitative investigation, we selected the number of commits performed in a time range as a measure for representing the productivity of a software development community. Even though we selected such a metric motivated by previous usage in past literature [57–59]—with encouraging results—we were aware that it could be the origin of incorrect results; indeed, productivity is a complex concept to define and measure, and using a single representation could not be sufficient during a research analysis.

For the reason mentioned above, we decided to conduct a qualitative study, and we asked participants to provide us with some examples of productivity measures that they consider attentive. As shown in Figure 4, participants consider the number of commits as a good metric to measure the productivity of a community. Therefore, we are confident that the quantitative and qualitative studies' results are reliable.

4.2.3. *Culture and Cultural Behaviors*

In the first question of each section, we asked participants to evaluate how many times they faced the described scenario in the past using a Likert scale from *Never* to *Always*. As shown in Figure 5, most participants recognized the different cultural dimensions inside their team. In particular, *Individualism vs. Collectivism Index* and *Indulgence vs. Restraint Index* were the most frequent (respectively, 40% and 50% claimed that they often experienced it), while *Power Distance Index* was less frequent (30% of participants claimed that they *never* experienced it and 33% that they experienced it *sometimes*).

Besides studying the frequency of experienced behaviors, we were also interested in whether practitioners perceive culture as the principal root of such behaviors. Specifically, our goal was to verify if practitioners perceive a connection between these behaviors and the conjecture of Hofstede that relates them to cultural factors. For such a goal, we asked participants to what extent the individuals' culture could influence the emergence of specific behaviors. As shown in Figure 6, most participants related the behaviors to cultural factors. In fact, in most cases, people do not answer *Not at all* or *Very little* but *Somewhat* and *To a great extent*. The only exceptions are for UAI, for which *Not at all* is the most selected (13%), and LTO, for which *Not at all* and *Very little* are the most selected ones (43% and 33%). Based on our result, we can conclude that the survey confirmed the relevance of cultural dimensions in software development

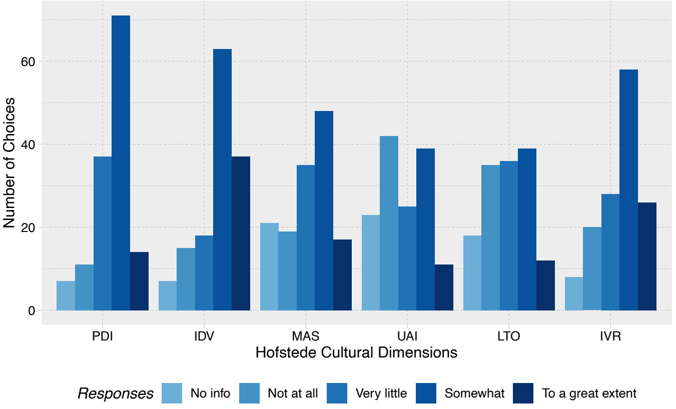

Figure 6: Relation between culture and behaviors.

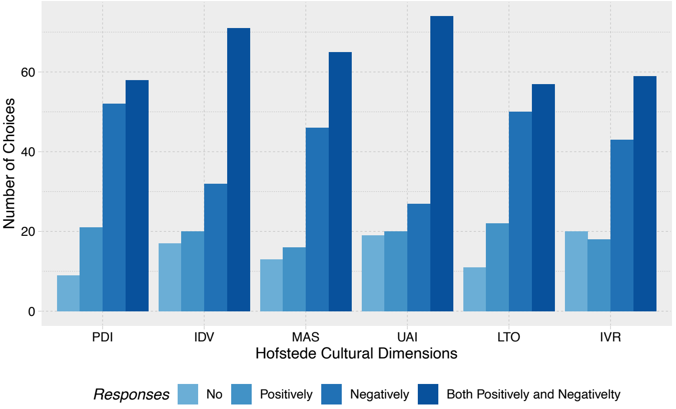

Figure 7: Impact of cultural behaviors on productivity.

communities and the relationship between belonging to different cultures and having certain behaviors.

4.2.4. *Impact of dispersion metrics on productivity*

As stated in Section 2.1, one cultural dimension is defined by two opposite behaviors, so in the context of the survey, we asked our participants whether the presence of both

i) does not affect productivity,

ii) influences it positively (by increasing it),

iii) influences it negatively (by decreasing it), or

iv) influences it both positively and negatively.

In general, as reported in Figure 7, practitioners perceive all the contrast between behaviors as impactful for productivity (as demonstrated by the low percentages of *No* answers). In fact, the higher value for *No* is nine responses—i.e., 7% of respondents. Furthermore, they generally consider such an impact negatively or both negatively and positively. The only differences in this are for *Individualism vs. Collectivism* and *Uncertainty Avoidance Index* behaviors, in which we have a more significant number of responses for *Both negatively and positively* (respectively, 47% and 49%).

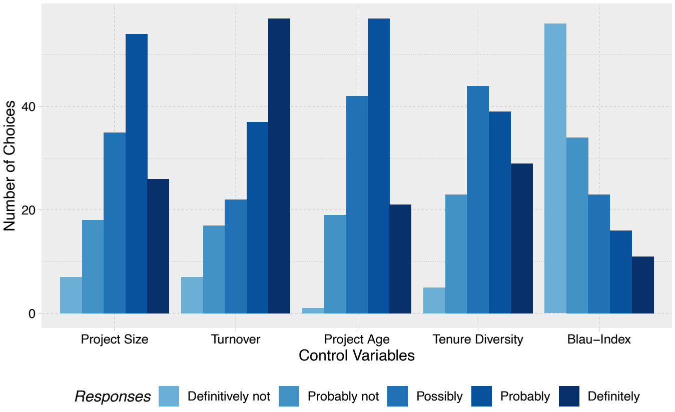

Figure 8: Impact of control variables on productivity.

Regarding geographical dispersion, participants confirmed the statistical model results, assessing that physical distance is a crucial factor for the productivity of individuals. Indeed, one of the participants assessed that *"When people work remotely, and communication is not optimal, it is often necessary to repeat and meet several times before what needs to be done is done."*.

The survey confirmed the quantitative study results. Indeed *PDI*, *LTO*, and *IDV* are perceived as impactful. However, also the other behaviors are perceived as impactful by participants.

4.2.5. *Impact of control variables*

Regarding the perceived impact of control variables as shown in Figure 8, most of them are perceived as impactful on software development productivity—i.e., Team Size, Turnover, Project Age, Tenure diversity, and Blau-Index. The only exception to this is Blau gender, which is perceived as not impactful.

Qualitative study: summary of the results.

Practitioners perceive cultural and geographical dispersion as impactful for the productivity of a software team. Moreover, in most cases, such an impact is double-fashioned, i.e., both negative and positive; this is because having a high level of cultural dispersion could—in general—lead to more extended discussions and lower team productivity. Therefore, correctly managing cultural differences is essential to bring out the best in one's developers.

Theoretical Saturation.

Both the qualitative and the quantitative study converges to the same high-level result: dispersion metrics impact the productivity of a development community greatly. Moreover, the impact of cultural behaviors on the productivity of a software community can be categorized and analyzed using cultural dimensions, leading to the possibility for managers and team leaders to make more precise decisions. Indeed, cultural dispersion impacts productivity differently based on original Hofstede's dimension. Furthermore, the two studies reveal that the influence of dispersion metrics is positive and negative. The dispersion in the team led to extended discussion resulting in (i) reducing productivity due to the time lost in communicating but (ii) increasing it because the discussion helped practitioners identify crucial tasks to perform. Using our findings, managers could find a converging point easier and reduce productivity loss by knowing the contrast point—i.e., the cultural dimension and corresponding behavior—enhancing the positive impact. For example, by understanding that the team is dispersed in terms of Power Distance Index Dispersion—i.e., there are different attitudes in how relating to the distribution of the power—managers could implement ad hoc solutions—e.g., changing the decision-making process or implementing a voting system—for reducing the negative impact of such dispersion and enhancing the positive aspects.

5. Discussion and Implications

The insights of our study shed light on the role and the impact of cultural dimensions across software organizational structures revealing several lessons for software project management and community shepherding.

The role of culture as a factor influencing productivity in software development teams has yet to deepen. In this work, we tried to fill this gap by focusing on cultural and geographical dispersion. The following section discusses the final considerations based on the results achieved in Section 4.

***Productivity Metrics: Quantitative and Qualitative*.** As already said, productivity is a complex concept to define and measure. Indeed, a large plethora of work provided different interpretations and measurements [57–59]. From such heterogeneity of materials on the matter, conducting a study that captures all the aspects of productivity comes to be a complex and tricky challenge. Nevertheless, managers can only manage what they can measure, so it is mandatory to study the phenomena and provide new insights. Aware of the complex side of productivity, we tried to address it by combining quantitative and qualitative representations in our studies. The former—i.e., the number of commits—allowed us to be more "objective" in our findings; the latter—i.e., self-assessed productivity—allowed us to capture more aspects and indicators of productivity. Ultimately, the two studies reported similar results, providing a potential precedent for ulteriorly enhanced findings on the matter using similar methods.

***Individualism vs. Collectivism Dispersion*.** Regarding the influence of *IDVD* on productivity, both the qualitative and the quantitative studies showed that such a dispersion metric could positively influence the productivity of a software development community. As proof, we informally discuss these results with some practitioners with experience as managers in open-source projects from a large company. They highlighted that the presence of both individualistic and collectivistic people in the same team could increase productivity. Specifically, one of them reports that *"In my attempt to try to integrate an individualistic team member, I found that it was dragging on performance for the other individuals and me."*. This assertion could also justify the second predominant result from the survey, i.e., dispersion could also negatively affect the dependent variable—from a statistical point of view. Indeed, this situation can occur considering the example above, i.e., trying to enforce a paradigm change in the communication and collaboration pattern of the individualistic person.

⚠ We could draw that supporting individualistic behavior leads to the emergence of "positive lone wolves", thus improving the community members' productivity.

Such a fact was also confirmed by various participants of the survey: for example, one of them—asked about the root causes of problems originated by such a dispersion—reported this: *"It is okay for people to prefer to work alone as long as they do the work and show up to all meetings to collaborate with the team and show the work they have done."*.

***Long vs. Short Term Orientation and Power Distance Index Dispersion*.** Both *LTOD* and *PDID* negatively affect the productivity of a development community. These metrics reflect similar behaviors: indeed, *LTOD* indicates a contrast between people who tend to resist change—e.g., try out new programming languages or technologies—in contrast to others who are more flexible; *PDID*, indicates a difference between individuals who demand to equalize the distribution of the power between all team members, in contrast to others who want to follow a rigidly hierarchical organization.

⚠ In software development, these metrics can lengthen the time required to make a decision, thus possibly decreasing productivity when developing software artifacts or making decisions.

As a confirmation of this, one of the participants in the survey reported that *"When the difference in behaviors is from the team managing the processes, this can lead to delays whilst decisions are made."*.

⚠ Moreover, an ulterior point could be that the resentment arising from the contrast between individuals could lead to the formation of *Organizational Silos*—i.e., the situation in which different clusters of individuals in the same team arise and the communication between such "silos" is poor [82, 103]—, thus decreasing the communication and increasing the time to perform actions.

As a confirmation of this, one of the respondents to the questionnaire reports that *"Less collaboration occurs because people resent colleagues and don't want to share credit, so things get done slower."*.

Despite the negative influence that LTOD and PDID have on productivity, it is crucial to remember that low productivity does not necessarily mean low quality. In other words, more extended discussions—other than lowering developers' productivity—could lead to better identification of problems and their solutions, resulting in better quality—e.g., a better understanding of requirements. The critical aspect here is to prevent discussions from escalating into unnecessarily lengthy communications or, at worst, toxic arguments that could negatively impact the community's productivity. Related to this, we have observed a strong correlation between cooperation and collaboration issues and the concept of community smells [103, 104], which participants also mentioned, i.e., organizational silos. In our previous research [105], we have presented strategies to mitigate some of these smells. We believe that these strategies can be valuable in addressing the problems mentioned above and enhancing productivity.

One approach is facilitating mentoring activities to support team members in adapting to changes. This can involve organizing workshops, providing mentoring opportunities, or granting access to educational resources. By offering guidance and assistance, we can help individuals overcome their resistance to change and navigate new approaches more effectively. In addition, emphasizing clear communication is essential. It ensures the team comprehends the proposed changes' vision, goals, and rationale. When the reasoning is effectively communicated, it fosters understanding, reduces resistance, and encourages buy-in from the team members. Lastly, engaging in cohesion exercises with team leaders can play a vital role in promoting a positive attitude toward change and adaptation. Team leaders set an example for others to follow by demonstrating a willingness to embrace change. Being open to feedback and displaying a constructive mindset creates an environment where change is embraced as an opportunity for growth.

Control Variables. Our study confirms previous findings [15, 62]: indeed, socio-technical metrics [76, 106], e.g., turnover and tenure diversity, are strongly correlated with the productivity of a development community. For example, *centrality*—the degree to which a community is divided into subcommunities [81]—positively influences the dependent variable. The reason could be that, with the increasing number of developers, modularising the team could lead to better micromanagement, improving the community's productivity.

In addition, as we might expect, the *number of committers* and the *age of the project* also affect productivity. Concerning the first variable, it probably depends on the way chosen to represent productivity in this study (the number of commits): more committers likely lead to more commits over time. Regarding the second variable, the project's age negatively impacts the community's productivity. This could be because some open-source communities tend to die over time, mainly if some core contributors migrate to other teams.

As a final note, the survey's participants found *Blau-Gender*—i.e., the degree to which a community is different in terms of biological gender composition—as not relevant when analyzing the productivity of development communities. Both the analysis in this study confirmed these results. For example, Catolino et al. [27] found that gender diversity is perceived as being less important than experience or team size to mitigate the emergence of communication and collaboration issues—represented using community smells. Nevertheless, other quantitative investigations demonstrated that gender diversity could positively impact product and process metrics in software development [25, 26, 82]. Undoubtedly, ulterior research is needed to assess the influence of gender diversity in software development communities, maybe introducing novel methodologies for conducting investigations.

Benefits for tool vendors and developers. Nowadays, managing software development teams and conducting software projects is an expensive effort—both in terms of money, time, and human resources. Moreover, such an effort ulteriorly arises in the context of distributed teams characterized by a large plethora of factors, first of all, the extreme diversity of stakeholders in terms of behaviors and beliefs [10, 11]. Furthermore, developing guidelines, methods, and tools to help practitioners is complex due to the—at least apparently—abstractive nature of such social aspects. Nevertheless, our findings reveal that quantitative frameworks for measuring dispersion in development teams could be effective and can constitute a foundational step in developing recommendation systems for managers. Indeed, as a basilar example, using only the nationalities of team members—without any private information on the actual individuals—could allow managers to compute the dispersion rates and improve the estimation of productivity effort required for the project, constituting—if performed automatically—a worthwhile contribution.

> ⚠ Recommendation systems designers and developers should take the opportunity provided by quantitative representation of social aspects by developing tools to support practitioners in using and managing such factors. For example, computing a development team's dispersion metrics could contribute to the management process of software development projects.

We should care about software community health. These results shed light on an important aspect often underestimated by managers. Indeed, the health status of a community—its behaviors and diversity—needs to be carefully monitored when dealing with software development since it can affect the quantity (productivity) and quality of what it produces. Our conclusions can represent the first step toward better characterizing a software community in terms of "health".

6. Threats to Validity

This section illustrates the threats to the validity of the study and how we mitigated them. Other than using the guidelines provided for qualitative studies [91], we identified and organized the threats using the well-known framework proposed by Wohlin et al. [65, 107].

6.1. Threats to Construct Validity

Threats in this category refer to the relationship between hypothesis and observations and are mainly due to imprecision in performed measurements [65].

The first threat concerns the usage of Hofstede [28] for characterizing cultural dimensions. Although a few researchers raised concerns about this framework [47–49], Venkateswaran and Ojha [35] showed how the framework is the most efficient way to represent the complex world of cultural attitudes [32–34, 53].

A further concern regards the usage of the metric chosen for measuring productivity. Specifically, we used the number of commits for the quantitative investigation, while for the qualitative one, we relied on self-assessed productivity. This difference could introduce some imprecision and threats to the theoretical saturation of the studies. To mitigate this, in the first section of the survey, we asked participants to provide us with some examples of productivity measures that they consider attentive. As shown in Figure 4, participants consider the number of commits as a good metric to measure the productivity of a community.

As for the cultural and geographical dispersions metrics, we use the standard deviation since these metrics can be unreliable in the case of skewed measures. We applied the well-known Shapiro–Wilk test [108] to verify the normality of the data. Moreover, these metrics have already been adopted in a few studies in the context of software development [68, 69, 82]. Nevertheless, we encourage replication of the study using different measures.

Another threat is related to the dataset chosen to conduct our study. To address such a threat, we relied on a dataset already used and tested in similar studies [26, 66, 67].

Regarding the threat concerning the survey design, we followed the guidelines provided by Kitchenham and Pfleeger [91] and Andrews et al. [98] to define clear and explicit questions in the survey that could allow participants to get into the survey correctly. We also included open questions to let participants express themselves freely, without restriction. Finally, before sending the survey, we conducted a pilot study with four developers that reported possible biases and flaws we fixed before releasing the survey.

Our final concern pertains to the potential existence of subcultures within a larger cultural group. According to social science literature, individuals who share common backgrounds and beliefs might display minor variations in behavior, leading to the identification of sub-groups within the same cultural category [83, 84]. In our study, we utilized Hofstede's framework as a means to assess culture, which primarily focuses on defining culture at the country level, possibly overlooking subcultures. However, it should be acknowledged that Hofstede considered the presence of subcultures while designing the survey and assigning values to each country. Therefore, we are confident that the different sub-cultures behaviors are represented inside the used framework.

6.2. Threats to Internal Validity

Threats in this category are concerned with the possibility that the independent variable is affected by casualty factors without the researcher's knowledge [65].

The main threat is how we recruited our participants and their capacity to report about culturally dispersed teams. For the survey, we relied on voluntary participation through an online instrument like Prolific. In particular, this platform allows access to a pool of participants samples based on specific characteristics and backgrounds, e.g., computer science, working as developers. In our case, we looked for people with (1) a background in computer science, (2) experience in information services and data processing, and (3) experience in distributed teams. Furthermore, we tried to balance the number of participants based on gender as possible.

The participant's capacity to report about culturally dispersed teams is a second critical threat. Prolific allows us to distribute the survey to globally distributed participants (more information on this is in the responses to the last section of the survey, in the online appendix [37]). For such a reason—combined with a large number of participants—we are confident that the target audience has been represented. Nevertheless, we encourage replications of our work to strengthen our findings and their generalizability.

6.3. Threats to Conclusion Validity

Threats in this category are concerned with the ability to draw correct conclusions about relations between treatments and outcomes of an experiment [65].

The first threat concerns the statistical model selected for our study. We used a *mixed-effect model* [64, 84] to manage the multiple time windows for each project, thus capturing information within the same group. Additionally, we used *vif* for dealing with multicollinearity [85], and ANOVA test [86] for checking the significance of the results. Finally, to avoid omitting additional factors influencing a team's productivity, we included some socio-technical control factors identified by previous literature [14–16, 58, 62], e.g., socio-technical congruence.

A possible threat concerns the choice of Prolific Platform that involves the use of an incentive for people who do the survey, thus possibly affecting the results of our study. In our case, the participants involved in the survey obtained around 1$ for participating, thus allowing us to collect several answers. However, (1) the participation was voluntary, (2) we could collect various opinions from people with different experiences, thus increasing the reliability of our results, and (3) we asked to report their experience according to particular situations without influencing their answers. We released all the material used for this study to enable the verifiability of the conclusions described in our paper [37].

6.4. Threats to External Validity

Threats in this category are concerned with the generalizability of the results [65].

The main threat relates to the generalizability of the results. We use a dataset [66] containing information about big open-source projects on GitHub, with a large number of contributors.

Nevertheless, we plan to extend the number of systems and perform some qualitative studies, e.g., focus groups, surveys, and interviews, to strengthen the results as a future agenda. Moreover, we adopted a mixed-method approach, combining the results of a previously performed quantitative study with a large-scale survey. Although our numbers align with the SE research community's studies, we know how results strongly correlate with our sample, so replications are part of our agenda.

7. Conclusions

This study presents a qualitative extension of a quantitative empirical study [31] that investigates the relationship between the cultural and geographical dispersion of a community—i.e., the degree to which a community is formed by individuals growing up in and coming from different places globally—and its productivity, consider the number of commits in a specific range of time. The research question we wanted to answer was *"To what extent do cultural and geographical dispersion influence team's productivity?"*.

Our findings demonstrated that dispersion metrics impact productivity both positively and negatively. Indeed, how managers and leaders approach the differences between the cultural background of individuals is a crucial factor in determining their impact on the team. As an example of this, in the case of individualistic people collaborating with more collaboration-oriented ones, trying to integrate them into the team could lead to a useless and ineffective effort. Moreover, we found that the quantitative representations of social aspects are now mature to be exploited by practitioners. Developing tools and recommendation systems to track such metrics and visualize them to practitioners is an opportunity that should be considered.

As a future agenda, we plan to extend the generalizability of our results by adopting various productivity measurements. Moreover, we plan to develop instruments—e.g., conversational agents—aiming to perform a technology transfer from the research field to the practitioners one, making our findings easily usable by practitioners.

Acknowledgments

The authors would like to thank the participants of our study for the time and feedback provided. The authors would like to thank the two anonymous Reviewers for their insightful comments on our paper. Fabio is partially supported by the Swiss National Science Foundation - SNF Project No. PZ00P2_186090. Furthermore, this work has been partially supported by the EMELIOT national research project, which has been funded by the MUR under the PRIN 2020 program (Contract No. 2020W3A5FY).

References

[1] W. Scacchi, Understanding software productivity, in: Software Engineering and Knowledge Engineering: Trends for the next decade, World Scientific, 1995, pp. 273–316.

[2] W. N. Behutiye, P. Rodríguez, M. Oivo, A. Tosun, Analyzing the concept of technical debt in the context of agile software development: A systematic literature review, Information and Software Technology 82 (2017) 139–158.

[3] E. S. Cardozo, J. B. F. Araújo Neto, A. Barza, A. C. C. França, F. Q. da Silva, Scrum and productivity in software projects: a systematic literature review, in: 14th International Conference on Evaluation and Assessment in Software Engineering (EASE), 2010, pp. 1–4.

[4] S. Wagner, M. Ruhe, A systematic review of productivity factors in software development, arXiv preprint arXiv:1801.06475 (2018).

[5] E. Oliveira, T. Conte, M. Cristo, N. Valentim, Influence factors in software productivity—a tertiary literature review, International Journal of Software Engineering and Knowledge Engineering 28 (2018) 1795–1810.

[6] K. Petersen, Measuring and predicting software productivity: A systematic map and review, Information and Software Technology 53 (2011) 317–343.

[7] D. Girardi, F. Lanubile, N. Novielli, A. Serebrenik, Emotions and perceived productivity of software developers at the workplace, IEEE Transactions on Software Engineering 48 (2021) 3326–3341.

[8] I. Richardson, V. Casey, J. Burton, F. McCaffery, Global Software Engineering: A Software Process Approach, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 35–56. URL: https://doi.org/10.1007/978-3-642-10294-3_2. doi:10.1007/978-3-642-10294-3_2.

[9] S. Cherry, P. N. Robillard, 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, 2004, pp. 48–52.

[10] J. D. Herbsleb, D. Moitra, Global software development, IEEE software 18 (2001) 16–20.

[11] A. Mockus, J. Herbsleb, Challenges of global software development, in: Proceedings seventh international software metrics symposium, IEEE, 2001, pp. 182–184.

[12] C. Elbert, Global software engineering: Distributed development, outsourcing, and supplier management., IEEE Computer Society Books (2010).

[13] V. Casey, I. Richardson, A structured approach to global software development, European systems and software process improvement and innovation, Dublin, Ireland (2008).

[14] E. Murphy-Hill, C. Jaspan, C. Sadowski, D. Shepherd, M. Phillips, C. Winter, A. Knight, E. Smith, M. Jorde, What predicts software developers' productivity?, IEEE Transactions on Software Engineering 47 (2019) 582–594.

[15] S. Wagner, M. Ruhe, A systematic review of productivity factors in software development, arXiv preprint arXiv:1801.06475 (2018).

[16] P. Mohagheghi, R. Conradi, Quality, productivity and economic benefits of software reuse: a review of industrial studies, Empirical Software Engineering 12 (2007) 471–516.

[17] B. W. Boehm, Clark, Horowitz, Brown, Reifer, Chulani, R. Madachy, B. Steece, Software Cost Estimation with Cocomo II, 1st ed., Prentice Hall PTR, USA, 2000.

[18] D. Graziotin, X. Wang, P. Abrahamsson, Do feelings matter? on the correlation of affects and the self-assessed productivity in software engineering, Journal of Software: Evolution and Process 27 (2015) 467–487.

[19] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, V. Filkov, Gender and tenure diversity in github teams, in: Proceedings of the 33rd annual ACM conference on human factors in computing systems, 2015, pp. 3789–3798.

[20] L. Machuca-Villegas, G. P. Gasca-Hurtado, L. M. Restrepo Tamayo, S. Morillo Puente, Social and human factor classification of influence in productivity in software development teams, in: Systems, Software and Services Process Improvement: 27th European Conference, EuroSPI 2020, Düsseldorf, Germany, September 9–11, 2020, Proceedings 27, Springer, 2020, pp. 717–729.

[21] L. Machuca-Villegas, G. P. Gasca-Hurtado, M. Muñoz, Measures related to social and human factors that influence productivity in software development teams, International Journal of Information Systems and Project Management 9 (2021) 43–67.

[22] L. Machuca-Villegas, G. P. Gasca-Hurtado, S. M. Puente, L. M. R. Tamayo, Perceptions of the human and social factors that influence the productivity of software development teams in colombia: A statistical analysis, Journal of Systems and Software 192 (2022) 111408.

[23] N. Gorla, Y. W. Lam, Who should work with whom? building effective software project teams, Communications of the ACM 47 (2004) 79–82.

[24] D. A. Tamburri, F. Palomba, R. Kazman, Exploring community smells in open-source: An automated approach, IEEE Transactions on Software Engineering 47 (2019) 630–652. doi:10.1109/TSE.2019.2901490.

[25] F. Palomba, D. Andrew Tamburri, F. Arcelli Fontana, R. Oliveto, A. Zaidman, A. Serebrenik, Beyond technical aspects: How do community smells influence the intensity of code smells?, IEEE Transactions on Software Engineering 47 (2021) 108–129. doi:10.1109/TSE.2018.2883603.

[26] G. Catolino, F. Palomba, D. A. Tamburri, A. Serebrenik, F. Ferrucci, 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, 2019, pp. 11–20.

[27] G. Catolino, F. Palomba, D. A. Tamburri, A. Serebrenik, F. Ferrucci, Gender diversity and community smells: insights from the trenches, IEEE Software 37 (2019) 10–16.

[28] G. Hofstede, G. J. Hofstede, M. Minkov, Cultures and organizations: Software of the mind, volume 2, Mcgraw-hill New York, 2005.

[29] G. Hofstede, Dimensionalizing cultures: The hofstede model in context, Online readings in psychology and culture 2 (2011) 2307–0919.

[30] P. M. Institute, A Guide to the Project Management Body of Knowledge, 7 ed., 2021.

[31] S. Lambiase, G. Catolino, F. Pecorelli, D. A. Tamburri, F. Palomba, W.-J. Van Den Heuvel, F. Ferrucci, “there and back again?” on the influence of software community dispersion over productivity, in: 2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2022, pp. 177–184.

[32] G. Borchers, The software engineering impacts of cultural factors on multi-cultural software development teams, in: 25th International Conference on Software Engineering, 2003. Proceedings., IEEE, 2003, pp. 540–545.

[33] V. Casey, Imparting the importance of culture to global software development, ACM inroads 1 (2011) 51–57.

[34] S. Abufardeh, K. Magel, 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, IEEE, 2010, pp. 133–138.

[35] R. T. Venkateswaran, A. K. Ojha, Abandon hofstede-based research? not yet! a perspective from the philosophy of the social sciences, Asia Pacific Business Review 25 (2019) 413–434.

[36] R. B. Johnson, A. J. Onwuegbuzie, L. A. Turner, Toward a definition of mixed methods research, Journal of mixed methods research 1 (2007) 112–133.

[37] S. Lambiase, G. Catolino, F. Pecorelli, D. A. Tamburri, F. Palomba, F. Ferrucci, W.-J. van den Heuvel, A mixed-method empirical investigation into the influence of software communities cultural and geographical dispersion over productivity – online appendix, 2022. URL: https://figshare.com/s/92e387417ca21196cdfe.

[38] M. Marinho, A. Luna, S. Beecham, Global software development: practices for cultural differences, in: International Conference on Product-Focused Software Process Improvement, Springer, 2018, pp. 299–317.

[39] V. Stray, N. B. Moe, 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.

[40] J. Noll, S. Beecham, I. Richardson, Global software development and collaboration: barriers and solutions, ACM inroads 1 (2011) 66–78.

[41] S. Deshpande, I. Richardson, V. Casey, S. Beecham, Culture in global software development - a weakness or strength?, in: 2010 5th IEEE International Conference on Global Software Engineering, 2010, pp. 67–76. doi:10.1109/ICGSE.2010.16.

[42] H. Shah, N. J. Nersessian, M. J. Harrold, W. Newstetter, 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, ICIC ’12, Association for Computing Machinery, New York, NY, USA, 2012, p. 77–86.

[43] R. Kreitner, A. Kinicki, M. Buelens, Organizational Behavior, first european edition ed. london uk ed., McGraw-Hill Publishing Company, 1999.

[44] A. Furnham, The psychology of behaviour at work: The individual in the organization, Psychology press, 2012.

[45] E. T. Hall, Beyond culture, Anchor, 1989.

[46] C. Hampden-Turner, F. Trompenaars, C. Hampden-Turner, Riding the waves of culture: Understanding diversity in global business, Hachette UK, 2020.

[47] K. H. Roberts, N. A. Boyacigiller, 3. cross-national organizational research: The grasp of the blind men, in: Societal Culture and Management, De Gruyter, 2012, pp. 51–69.

[48] G. Ailon, Mirror, mirror on the wall: Culture’s consequences in a value test of its own design, Academy of management review 33 (2008) 885–904.

[49] R. F. Baskerville, Hofstede never studied culture, Accounting, organizations and society 28 (2003) 1–14.

[50] P. Brewer, S. Venaik, On the misuse of national culture dimensions, International Marketing Review (2012).

[51] P. Brewer, S. Venaik, The ecological fallacy in national culture research, Organization Studies 35 (2014) 1063–1086.

[52] A. Sorge, Review of culture’s consequences: International differences in work-related values, Administrative Science Quarterly 28 (1983) 625–629.

[53] G. Hofstede, 50 years memory lane – developing cultural dimensions from ibm data, Software of the Mind 2.0 (2017).

[54] A. Darwish, A. Henryson, How do cultural characteristics and software engineering practices interplay? (2019). URL: https://gupea.ub.gu.se/handle/2077/62534.

[55] E. Murphy-Hill, C. Jaspan, C. Sadowski, D. Shepherd, M. Phillips, C. Winter, A. Knight, E. Smith, M. Jorde, What predicts software developers’ productivity?, IEEE Transactions on Software Engineering 47 (2021) 582–594. doi:10.1109/TSE.2019.2900308.

[56] C. Sadowski, T. Zimmermann, Rethinking productivity in software engineering, Springer Nature, 2019.

[57] A. Mockus, R. T. Fielding, J. D. Herbsleb, Two case studies of open source software development: Apache and mozilla, ACM Transactions on Software Engineering and Methodology (TOSEM) 11 (2002) 309–346.

[58] A. Hernández-López, R. Colomo-Palacios, Á. García-Crespo, Software engineering job productivity—a systematic review, International Journal of Software Engineering and Knowledge Engineering 23 (2013) 387–406.

[59] E. Oliveira, E. Fernandes, I. Steinmacher, M. Cristo, T. Conte, A. Garcia, Code and commit metrics of developer productivity: a study on team leaders perceptions, Empirical Software Engineering 25 (2020) 2519–2549.

[60] P. J. Adams, A. Capiluppi, C. Boldyreff, Coordination and productivity issues in free software: The role of brooks’ law, in: 2009 IEEE International Conference on Software Maintenance, IEEE, 2009, pp. 319–328.

[61] D. Sornette, T. Maillart, G. Ghezzi, How much is the whole really more than the sum of its parts? 1 1= 2.5: Superlinear productivity in collective group actions, Plos one 9 (2014) e103023.

[62] S. R. de Lemos Meira, E. A. Barros, G. S. de Aquino, M. J. C. Silva, A review of productivity factors and strategies on software development, in: 2010 fifth international conference on software engineering advances, IEEE, 2010, pp. 196–204.

[63] Y. W. Ramírez, D. A. Nembhard, Measuring knowledge worker productivity: A taxonomy, Journal of intellectual capital (2004).

[64] M. J. Lindstrom, D. M. Bates, Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data, Journal of the American Statistical Association 83 (1988) 1014–1022.

[65] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.

[66] S. Lambiase, G. Catolino, D. A. Tamburri, A. Serebrenik, F. Palomba, F. Ferrucci, Good fences make good neighbours? on the impact of cultural and geographical dispersion on community smells, in: 2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), ACM, 2022, p. to appear.

[67] B. Vasilescu, A. Serebrenik, V. Filkov, A data set for social diversity studies of github teams, in: 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, 2015, pp. 514–517. doi:10.1109/MSR.2015.77.

[68] D. A. Tamburri, F. Palomba, A. Serebrenik, A. Zaidman, Discovering

community patterns in open-source: a systematic approach and its evaluation, Empirical Software Engineering 24 (2019) 1369–1417.

[69] W. Li, C. Yang, C. Yang, An active crawler for discovering geospatial web services and their distribution pattern–a case study of ogc web map service, International Journal of Geographical Information Science 24 (2010) 1127–1147.

[70] L. Bao, T. Li, X. Xia, K. Zhu, H. Li, X. Yang, How does working from home affect developer productivity?—a case study of baidu during the covid-19 pandemic, Science China Information Sciences 65 (2022) 142102.

[71] N. Forsgren, M.-A. Storey, C. Maddila, T. Zimmermann, B. Houck, J. Butler, The space of developer productivity: There's more to it than you think., Queue 19 (2021) 20–48.

[72] G. Catolino, F. Palomba, D. A. Tamburri, A. Serebrenik, Understanding community smells variability: A statistical approach, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), IEEE, 2021, pp. 77–86.

[73] F. Palomba, D. A. Tamburri, Predicting the emergence of community smells using socio-technical metrics: a machine-learning approach, Journal of Systems and Software 171 (2021) 110847.

[74] S. MacCurtain, P. C. Flood, N. Ramamoorthy, M. A. West, J. F. Dawson, The top management team, reflexivity, knowledge sharing and new product performance: A study of the irish software industry, Creativity and Innovation Management 19 (2010) 219–232.

[75] P. M. Blau, Inequality and heterogeneity: A primitive theory of social structure, volume 7, Free Press New York, 1977.

[76] G. Valetto, M. Helander, K. Ehrlich, S. Chulani, M. Wegman, C. Williams, 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, 2007, pp. 25–25.

[77] L. Williams, R. R. Kessler, Pair programming illuminated, Addison-Wesley Professional, 2003.

[78] G. Avelino, L. Passos, A. Hora, M. T. Valente, A novel approach for estimating truck factors, in: 2016 IEEE 24th International Conference on Program Comprehension (ICPC), IEEE, 2016, pp. 1–10.

[79] M. Ferreira, M. T. Valente, K. Ferreira, A comparison of three algorithms for computing truck factors, in: 2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC), IEEE, 2017, pp. 207–217.

[80] G. Avelino, E. Constantinou, M. T. Valente, A. Serebrenik, 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, 2019, pp. 1–12. URL: https://doi.org/10.1109/ESEM.2019.8870181. doi:10.1109/ESEM.2019.8870181.

[81] J.-P. Hatala, J. George Lutta, Managing information sharing within an organizational setting: A social network perspective, Performance Improvement Quarterly 21 (2009) 5–33.

[82] S. Lambiase, G. Catolino, D. A. Tamburri, A. Serebrenik, F. Palomba, F. Ferrucci, Good fences make good neighbours? on the impact of cultural and geographical dispersion on community smells, in: Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society, ICSE-SEIS '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 67–78. URL: https://doi.org/10.1145/3510458.3513015. doi:10.1145/3510458.3513015.

[83] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, V. Filkov, Gender and Tenure Diversity in GitHub Teams, Association for Computing Machinery, New York, NY, USA, 2015, p. 3789–3798. URL: https://doi.org/10.1145/2702123.2702549.

[84] D. Bates, M. Mächler, B. Bolker, S. Walker, Fitting linear mixed-effects models using lme4, arXiv preprint arXiv:1406.5823 (2014).

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

[86] A. Cuevas, M. Febrero, R. Fraiman, An anova test for functional data, Computational statistics & data analysis 47 (2004) 111–122.

[87] K. P. Burnham, D. R. Anderson, Multimodel inference: understanding aic and bic in model selection, Sociological methods & research 33 (2004) 261–304.

[88] H. Akaike, Information theory and an extension of the maximum likelihood principle, in: Selected papers of hirotugu akaike, Springer, 1998, pp. 199–213.

[89] K. P. Burnham, D. R. Anderson, Multimodel inference: understanding aic and bic in model selection, Sociological methods & research 33 (2004) 261–304.

[90] A. N. Meyer, L. E. Barton, G. C. Murphy, T. Zimmermann, T. Fritz, The work life of developers: Activities, switches and perceived productivity, IEEE Transactions on Software Engineering 43 (2017) 1178–1193.

[91] B. A. Kitchenham, S. L. Pfleeger, Personal opinion surveys, in: Guide to advanced empirical software engineering, Springer, 2008, pp. 63–92.

[92] J. Finch, The vignette technique in survey research, Sociology 21 (1987) 105–114.

[93] C. Atzmüller, P. M. Steiner, Experimental vignette studies in survey research, Methodology (2010).

[94] A. W. Meade, S. B. Craig, Identifying careless responses in survey data., Psychological methods 17 (2012) 437.

[95] K. J. Hunt, N. Shlomo, J. Addington-Hall, Participant recruitment in sensitive surveys: a comparative trial of 'opt in'versus 'opt out'approaches, BMC Medical Research Methodology 13 (2013) 1–8.

[96] J. J. Heckman, Selection bias and self-selection, in: Econometrics, Springer, 1990, pp. 201–224.

[97] J. W. Sakshaug, A. Schmucker, F. Kreuter, M. P. Couper, E. Singer, Evaluating active (opt-in) and passive (opt-out) consent bias in the transfer of federal contact data to a third-party survey agency, Journal of Survey Statistics and Methodology 4 (2016) 382–416.

[98] D. Andrews, B. Nonnecke, J. Preece, Conducting research on the internet:: Online survey design, development and implementation guidelines (2007).

[99] B. Reid, M. Wagner, M. d'Amorim, C. Treude, Software engineering user study recruitment on prolific: An experience report, arXiv preprint arXiv:2201.05348 (2022).

[100] F. Ebert, A. Serebrenik, C. Treude, N. Novielli, F. Castor, On recruiting experienced github contributors for interviews and surveys on prolific, in: International Workshop on Recruiting Participants for Empirical Software Engineering, 2022.

[101] T. S. Flanigan, E. McFarlane, S. Cook, Conducting survey research among physicians and other medical professionals: a review of current literature, in: Proceedings of the Survey Research Methods Section, American Statistical Association, volume 1, 2008, pp. 4136–47.

[102] S. Cavanagh, Content analysis: concepts, methods and applications., Nurse researcher 4 (1997) 5–16.

[103] D. A. Tamburri, R. Kazman, H. Fahimi, The architect's role in community shepherding, IEEE Software 33 (2016) 70–79.

[104] D. A. Tamburri, P. Lago, H. v. Vliet, Organizational social structures for software engineering, ACM Computing Surveys (CSUR) 46 (2013) 1–35.

[105] G. Catolino, F. Palomba, D. A. Tamburri, A. Serebrenik, F. Ferrucci, 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, 2020, pp. 25–34.

[106] M. Cataldo, P. A. Wagstrom, J. D. Herbsleb, K. M. Carley, 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, 2006, pp. 353–362.

[107] C. Wohlin, M. Höst, K. Henningsson, Empirical research methods in software engineering, in: Empirical methods and studies in software engineering, Springer, 2003, pp. 7–23.

[108] S. S. Shapiro, M. B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (1965) 591–611.