"There and Back Again?" On the Influence of Software Community Dispersion Over Productivity

Stefano Lambiase*, Gemma Catolino†, Fabiano Pecorelli‡, Damian A. Tamburri†, Fabio Palomba*, Willem-Jan van den Heuvel†, Filomena Ferrucci*
*SeSa Lab – University of Salerno, Salerno, Italy
†Jheronimus Academy of Data Science – Tilburg University, 's-Hertogenbosch, Netherlands
‡CloudSEA.AI – Tampere University, Tampere, Finland

***Abstract*—Estimating and understanding productivity still represents a crucial task for researchers and practitioners. Researchers spent significant effort identifying the factors that influence software developers' productivity, providing several approaches for analyzing and predicting such a metric. Although different works focused on evaluating the impact of human factors on productivity, little is known about the influence of cultural/geographical diversity in software development communities. Indeed, in previous studies, researchers treated cultural aspects like an abstract concept without providing a quantitative representation. This work provides an empirical assessment of the relationship between *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—and its productivity. To reach our aim, we built a statistical model that contained product and socio-technical factors as independent variables to assess the correlation with productivity, i.e., the number of commits performed in a given time. Then, we ran our model considering data of 25 open-source communities on GitHub. Results of our study indicate that cultural and geographical dispersion impact productivity, thus encouraging managers and practitioners to consider such aspects during all the phases of the software development lifecycle.**

***Index Terms*—Global Software Engineering; Cultural Dispersion; Geographical Dispersion; Productivity; Software Organizational Structures; Empirical Studies.**

I. Introduction

Nowadays, software development is, even more, a globally distributed activity [1], [2], thus involving stakeholders from different places globally and, consequently, different cultures and backgrounds. Researchers in the context of *Global Software Engineering* (GSE) [3], [4]—i.e., the set of practices and guidelines aimed at managing software distributed teams—demonstrated that such heterogeneity in development communities leads to positive behavior [1], [5] but, if not well managed, it can lead to some issues, e.g., difficulties in communication and misunderstandings between team members [4], [6], [7]. Futherthermore, aspects related to developers' communication and interaction have been discovered to be related to the productivity of a software community [8]–[13].

Among the various aspects taken into account by GSE researchers, *culture*—the way people act and think [14]—is defined as one of the most complex factors with respect to several software metrics [15]–[18]. For defining culture, researchers started using the *Hofstede's Framework* [19], [20] to represent the culture of individuals. The framework consists of six behavioral characteristics—which can assume values from zero to one hundred—that can culturally characterize individuals according to their home country. The extensive research on productivity mentioned above showed how essential human aspects are in treating this complex topic. However, previous works did not focus the attention on cultural aspects from a concrete perspective.

With this study, we aimed to address the limitations mentioned above by conducting a quantitative empirical study 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.

To operationalize the *cultural dispersion* of a development community, we use the well-known Hofstede Framework [19], [20], already used in previous works in SE [21]–[24]. As for the *geographical dispersion*, we used the spherical distance between the community's members. Afterward, 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 key results of our study indicate that dispersion metrics influence developers' productivity both positively and negatively. For example, the presence of people coming from a more rigid culture in contrast to others who are more flexible could reduce the team's work rate. In addition, having team members who agree on the best way to work together—whether closely or in isolation—can positively impact productivity. To sum up, our article provides two contributions:

1) A regression model that analyzes the influence of two dispersion metrics—cultural and geographical dispersion—on the productivity of a development community;
2) An online replication package [25] publicly available to support replication and future work.

These results shed light on an important aspect often underestimated by managers. Indeed, the health status of a community—their behaviors and diversity—needs to be carefully monitored when dealing with software development since it can affect the quantity (productivity) and quality of what produces.

Regarding the structure of the paper, Section II describes the existing literature related to cultural factors in software engineering and productivity. In Section III, we present and outline the methodology of our research, and Section IV reports the results. Section V discusses the insights of the paper, and Section VI examines the threats to the validity and their mitigation. Finally, Section VII concludes the paper and provides plans for the future research agenda.

II. Background and Related Work

This section describes the background and related work in the context of our study.

A. Culture and Cultural Dimensions around Software Organizations

Hofstede defines *culture* as: *"The collective programming of the mind that distinguishes the members of one group or category of people from others."* [26]. Software development is becoming more globally distributed [1], [2], thus implying team members working from different places worldwide, or having different cultures [3], [17], [27], [28].

For operationalizing cultural factors and conducting research studies on the influence of these aspects, Hofstede [19], [20] defined his framework, i.e., **Hofstede's 6-D framework**. It is a set of six dimensions that assume values from zero to one hundred and which combination characterizes a specific country globally [19], [20], [29].

The six dimensions represent different aspects of human behaviors related to individuals' education. For instance, the dimension *Uncertainty Avoidance (UAI)* expresses the degree to which the members of a society are tolerant of uncertainty and ambiguity. People bought up in countries exhibiting a high level of *UAI* are not too keen on uncertainty: they tend to plan everything carefully and avoid uncertainty. Conversely, a low level of *UAI* indicates societies that maintain a more relaxed attitude and tend to plan less [19], [20]. A detailed description of Hofstede's dimensions can be found in different works [19], [20] and on the main website of the framework.[1]

The software engineering research community used Hofstede's dimensions in some works [21]–[23]. For instance, Abufardeh and Magel [23] demonstrated the impact of cultural factors on the Global Software Development process and software products, emphasizing the importance of further research. Finally, Borchers et al. [21] studied how teams consisting of people from three different cultures approached the software development process.

Although a few studies raised concerns about the framework [30]–[35], Venkateswaran and Ojha [36] showed how Hofstede's framework represents the most effective tool to represent a culture in several fields of application [21]–[23], [37]—e.g., management, law, politics, ethics, architecture, medicine, economics, and computer science.

B. Productivity Factors

Productivity represents a complex concept to define and measure in software development, especially when we need to measure it in the open-source field. Therefore, different definitions have been proposed to measure productivity [11], [38]–[40]. Despite this, the research community in software engineering agrees that productivity should be measured in terms of output produced in a given time given a specific input [39], [41].

For instance, 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 [39], [41].

Moving the attention to which factors influence productivity, we discovered that several researchers focused their attention on this aspect [8]–[13].

We can divide these studies based on the metrics found relevant. As for technical factors, i.e., product metrics and tools, Wagner and Ruhe [9] conducted a systematic review, showing how metrics such as code reuse, software size, and programming language highly impact developers' productivity. Moreover, Mohagheghi and Conradi [10] investigated the relationship between productivity and software reuse, showing positive results.

As for social factors—mainly related to people and their relations—Murphy-Hill et al. [8] surveyed practitioners, demonstrating that social factors—e.g., people's enthusiasm, peer support, and valuable feedback about job performance—significantly affect people's productivity. Additionally, Wagner and Ruhe [9] showed that social factors like corporate culture and working environment are essential aspects to be considered for enhancing software teams' productivity. Graziotin et al. [12] showed how valence and dominance dimensions in developers affect self-assessed productivity. Finally, Vasilescu et al. [13] demonstrated that gender and tenure diversity are positive and significant predictors of productivity using a statistical model.

Despite the various works in the past, we identified two main limitations. Even if the works of Murphy et al. [8] and Wagner et al. [9] are close to ours, we believe that they managed the context of culture as an abstract and high-level concept without operationalizing them. Indeed, we used Hofstede's 6-D framework [19], which was specifically defined to represent the cultural dimensions concretely. Finally, they performed a qualitative analysis [8], i.e., surveys and interviews. In contrast, we conducted a quantitative study, e.g., repository mining and statistical models construction, to enhance the generalizability of results.

III. Research Methodology

The *goal* of the study is to analyze whether cultural and geographical dispersion—i.e., the degree to which a community is formed by individuals with different cultural habits and working from different places—influence the productivity of a community, computed by counting the number of commits in a period of time [39]. The *purpose* is to provide new insights to allow practitioners to make more informed decisions based on their software development community. The *perspective* is of managers who are interested in effectively allocating

[1] Hofstede' dimensions: https://hi.hofstede-insights.com/national-culture

TABLE I: 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

resources, adhering to the project's requirements, or managing/monitoring complex organizational structures.
The study revolves around the following research question:

> **RQ**: To what extent do cultural and geographical dispersion influence teams' productivity?

To address our **RQ**, we constructed a statistical model able to assess whether the phenomenon of cultural and geographical dispersion relates to the productivity of open source development communities. We employed the guidelines by Wohlin et al. [42] and we followed the *ACM/SIGSOFT Empirical Standards*.[2] In particular, we used the *"General Standard"* and *"Data Science"* guidelines.

A. Data Collection

To conduct our study, we used the dataset already available from our previous study [24] 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 I reports the list of the projects and the number of windows considered for each of them in our study.

In the context of our study, the most valuable metrics—contained in this dataset—are the ones describing the cultural and geographical dispersion of software communities. As for the cultural dispersion, the dataset contains six cultural metrics that can assume values from zero to fifty. Each metric corresponds to the standard deviation of the set containing the community members' value for one of the six dimensions of Hofstede [19]. As for the geographical dispersion, the dataset provides the standard deviation of the spherical distances (in miles) between each community member as a metric—computed using the GEOPY[3] library. To calculate both the dispersion metrics, we used the original country of each developer in the development communities—already provided in the original dataset [43], [44]. Further details are provided in the following section.

B. RQ - Building a Statistical Model

To address our **RQ**, we defined a statistical linear regression model relating a development community's cultural and geographical dispersion to the productivity expressed in terms of the number of commits. Figure 1 summarizes our research methodology and the steps followed to build the model—explained in the following subsections.

1) Independent Variables: In the context of our study, we considered the following factors.

Cultural Dispersion. The *cultural dispersion* of a development community indicates "how such a community is formed by developers coming from different cultural behaviors" [24], [45]. To operationalize this concept, we used Hofstede's framework [19]. Specifically, we use six metrics, one for each dimension in the framework. Each of their values corresponds to the standard deviation of the set containing the community members' values. The metrics are reported in the following—their description relies on our previous study [24]:

- **PDID**: *Power Distance Index Dispersion* indicates how community members tend to have a different idea on hierarchical structure and power division.
- **IDVD**: *Individualism vs. Collectivism Dispersion* indicates how much community members tend to have different ideas regarding forming groups and sharing success.
- **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.

[2]Available at: https://github.com/acmsigsoft/EmpiricalStandards. We followed the *"General Standard"* and *"Data Science"* definitions and guidelines.

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

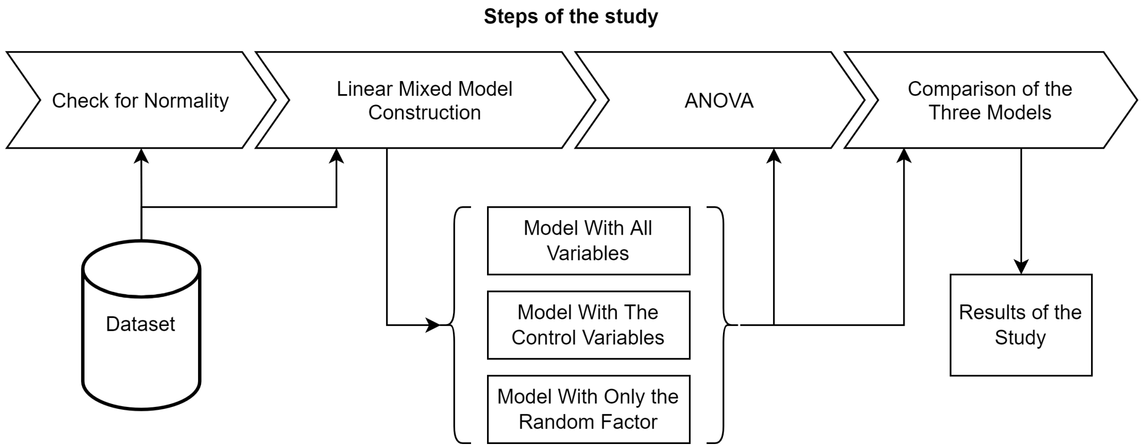

Fig. 1: Overview of our research methodology.

- **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.

Geographical Dispersion (GeoD) indicates "how a community is formed by developers working from different places around the world". It is defined as the standard deviation of the set of physical distances between each community member.

It is important to note that the metrics presented have been calculated for each time window since the project community members can change over time.

2) Response 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 [39], [41] as a reference measurement. We relied on this metric since it has been widely used when dealing with productivity operationalization [39]–[41].

3) Control Variables: When constructing a statistical model, it is essential to consider that beyond independent variables, other variables can affect the phenomenon analyzed, as demonstrated in the literature [12], [13], [24], [43], [46], [47]. 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 [48], thus possibly affecting productivity [13]. 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 project median 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 [49] 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 [50] 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 fail [51]–[54].
- **Centrality**: It is defined as the strength of a community, and it is based on modularity measures [55]. A value over 0.3 means that the community is highly modular, thus clearly distinguishing the sub-communities present 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 [12], [13], [24], [43], [46], [47].

4) Statistical Model Construction: Since the dataset provided by Lambiase et al. [24] consists of multiple temporal windows for each project analyzed—we constructed a *linear mixed model* able to capture measurements from within the same group (i.e., within the same project) as a random effect [56]. In the context of our study, we employed the time window as a random effect and the rest of the variables mentioned above as fixed effects. In particular, we relied on the functions `lmer` and `lmer.test` available in the `R` package `lme4` [57]. During our experiment, we faced the problem of multi-collinearity [58], 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 [58]. Finally, to strengthen our results, we computed the effect sizes of the coefficients using the ANOVA statistical test [59]. Variables are considered significant if they are statistically significant, i.e., the p-value is less than 0.05. For the sake of results reliability, we constructed two baseline statistical models: the first one containing all the control variables and the random effect. We compared the models with the corresponding baselines through the *AIC* (*Akaike information criterion*) and *BIC* (*Bayesian information criterion*) [60], [61] estimators. They are widely used for model selection criteria, assessing the quality of the prediction [62]. Indeed, *AIC* and *BIC* 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. The comparison with the first model allows us to understand how the control variables, without the independent ones, influence the productivity—expressed in terms of # of commits. The comparison with the second model whether the obtained results reflected the random effect instead.

IV. Analysis of the Results

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

Table II reports the details regarding the statistical models. The first model (see the column "All Variables") reports the results achieved for the model with both confounding factors and independent variables. As revealed, the *number of committers* and the *centrality of the community* seem to be excellent estimates of productivity. In addition, the *project age* and the *number of females* in the team significantly impact the dependent variable.

Regarding 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, dispersion metrics seem to be good predictors for the dependent variable and are worth considering in its estimation.

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

TABLE II: Results achieved.

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 III: AIC and BIC for the three models.

Metric	All Variables	Conf. Variables	Random
Akaike Information Criterion	502	536	617
Bayesian Information Criterion	572	584	626

To provide a preliminary interpretation of our results—similar to our previous study [24]—cultural and geographical dispersion influence the productivity of a development community both in a positive and negative fashion. As proof of this, *Individualism vs. Collectivism Dispersion*, *Indulgence vs Restraint Dispersion*, and *Geographical Dispersion* impact the dependent variable positively, while *Power Distance Index Dispersion* and *Long Term Orientation Dispersion* negatively.

Table III reports 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 variables and the random. Therefore, adding the independent variables contributes to explaining the response variable better.

Summary of the results.

☞ Our study confirmed how socio-technical metrics could significantly impact the productivity of a software development team. In addition, the study revealed that cultural differences and geographic dispersion among developers are essential factors in predicting community productivity.

V. Discussion and Implications

The role of culture as a factor influencing productivity in software development teams has not yet deepened. In this work, we tried to fill this gap by focusing on the concept of cultural and geographical dispersion. The following section discusses the final considerations based on the results achieved in Section IV.

Individualism vs Collectivism Dispersion. On the influence of *IDVD* on productivity, we noticed that this metric could positively affect it. As a confirmation, we informally discuss these results with some practitioners with experience as managers in open-source projects from a large company. They told us that the presence of both individualistic and collectivistic people—in the same team—could lead to increase productivity. Specifically, one of them reports *"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."*. We could draw that supporting individualistic behavior leads to the emergence of "positive lone wolves", thus improving the community members' productivity. Nevertheless, we plan further investigation based on such results.

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 contrast 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 lead to the lengthening of the time required to make a decision, thus possibly decreasing productivity when developing software artifacts. Such consideration can partially explain the observed correlation.

Control Variables. Our study confirms previous findings [9], [63]: indeed, socio-technical metrics [50], [64], e.g., *centrality*, are strongly correlated with the productivity of a development community. Specifically, *centrality*—the degree to which a community is divided into sub-communities [55]—positively influences the dependent variable. The reason could be that, with the increasing number of developers, modularising the team could lead to better micro-management, increasing the productivity of the entire community.

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.

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—their behaviors and diversity—needs to be carefully monitored when dealing with software development since it can affect the quantity (productivity) and quality of what produces. Our conclusions can represent the first step toward better characterizing a software community in terms of "health".

VI. Threats to Validity

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

A. 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 regards the use of standard deviation as a summary metric to represent cultural and geographical dispersion since these metrics can be unreliable in the case of skewed measures. For this reason, we applied the well-known Shapiro–Wilk test [66] for verifying the normality of the data.

The second threat is related to the dataset chosen to conduct our study. We relied on a dataset already used and tested in similar studies [24], [43], [44].

Another threat concerns the metric used for representing productivity. We operationalized it using the number of commits. Although various alternatives have been proposed by the research community [11], [40], [41], other studies [39], [41] demonstrated that the number of commits best suits the measure of productivity. Nevertheless, we plan to replicate our study using different types of reference metrics.

Finally, although the usage of Hofstede—to represent cultural aspects—has been criticized [30]–[32], Venkateswaran and Ojha [36] demonstrated that, currently, the framework is the most efficient way to characterize culture.

B. Threats to Conclusion Validity

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

The first threat concerns the statistical model selected for our study. We used a *mixed-effect model* [56], [57] to manage the multiple time windows for each project, thus capturing information within the same group.

Additionally, we used *vif* for dealing with multicollinearity [58], and ANOVA test [59] for checking the significance of the results.

Finally, to avoid the risk of omitting additional factors able to influence a team's productivity, we included some socio-technical control factors identified by previous literature [8]–[10], [40], [63], e.g., socio-technical congruence.

C. 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 [24] containing information about big open-source projects on GitHub, with 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.

VII. CONCLUSIONS

Our study aimed to understand the correlation between *cultural and geographical dispersion* and the productivity of a software development community. We found that the dispersion metrics impact productivity both positively and negatively. For instance, the presence in the same team of people who tend to resist change in contrast to others who are more flexible could reduce productivity in favor of an increased time allocated for communication activities. The study provides the following contributions:

1) A regression model that analyzes the influence of two dispersion metrics—cultural and geographical dispersion—on the productivity of a development community;
2) An online replication package [25] publicly available to support replication and future work.

As a future agenda, we plan to enhance the generalizability of the obtained results in two ways:

- By conducting qualitative studies—e.g., surveys, interviews, and focus groups;
- By replicating our study using different datasets and metrics to represent a development community's productivity.

Furthermore, we intend to study the correlation between the dispersion metrics and the *sustainability of a development community* [67], [68]—defined as the ability of a community to meet its own needs and ensure resources for future generations of developers—that has been demonstrated to relate with productivity [67], [68].

REFERENCES

[1] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE software*, vol. 18, no. 2, pp. 16–20, 2001.

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

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

[4] S. Cherry and 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.

[5] J. Portillo-Rodríguez, A. Vizcaino, M. Piattini, and S. Beecham, "Using agents to manage socio-technical congruence in a global software engineering project," *Information Sciences*, vol. 264, pp. 230–259, 2014.

[6] C. Elbert, "Global software engineering: Distributed development, outsourcing, and supplier management." *IEEE Computer Society Books*, 2010.

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

[8] E. Murphy-Hill, C. Jaspan, C. Sadowski, D. Shepherd, M. Phillips, C. Winter, A. Knight, E. Smith, and M. Jorde, "What predicts software developers' productivity?" *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 582–594, 2019.

[9] S. Wagner and M. Ruhe, "A systematic review of productivity factors in software development," *arXiv preprint arXiv:1801.06475*, 2018.

[10] P. Mohagheghi and R. Conradi, "Quality, productivity and economic benefits of software reuse: a review of industrial studies," *Empirical Software Engineering*, vol. 12, no. 5, pp. 471–516, 2007.

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

[12] D. Graziotin, X. Wang, and P. Abrahamsson, "Do feelings matter? on the correlation of affects and the self-assessed productivity in software engineering," *Journal of Software: Evolution and Process*, vol. 27, no. 7, pp. 467–487, 2015.

[13] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and 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.

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

[15] J. S. Olson and G. M. Olson, "Culture surprises in remote software development teams: When in rome doesn't help when your team crosses time zones, and your deadline doesn't." *Queue*, vol. 1, no. 9, pp. 52–59, 2003.

[16] G. Calikli, A. Bener, and B. Arslan, "An analysis of the effects of company culture, education and experience on confirmation bias levels of software developers and testers," in *2010 ACM/IEEE 32nd International Conference on Software Engineering*, vol. 2. IEEE, 2010, pp. 187–190.

[17] S. Deshpande, I. Richardson, V. Casey, and 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.

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

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

[20] G. Hofstede, "Dimensionalizing cultures: The hofstede model in context," *Online readings in psychology and culture*, vol. 2, no. 1, pp. 2307–0919, 2011.

[21] 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.

[22] V. Casey, "Imparting the importance of culture to global software development," *ACM inroads*, vol. 1, no. 3, pp. 51–57, 2011.

[23] S. Abufardeh and 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*, 2010, pp. 133–138.

[24] S. Lambiase, G. Catolino, D. A. Tamburri, A. Serebrenik, F. Palomba, and 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.

[25] S. Lambiase, G. Catolino, F. Pecorelli, D. A. Tamburri, F. Palomba, F. Ferrucci, and W.-J. van den Heuvel, "*"There and Back Again?"* on the influence of software community dispersion over productivity — online appendix," 2022. [Online]. Available: https://figshare.com/articles/dataset/_There_and_Back_Again_On_the_Influence_of_Software_Community_Dispersion_Over_Productivity_Online_Appendix/19576294

[26] G. Hofstede, *Culture's consequences: International differences in work-related values*. sage, 1984, vol. 5.

[27] H. Shah, N. J. Nersessian, M. J. Harrold, and 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*, ser. ICIC '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 77–86. [Online]. Available: https://doi.org/10.1145/2160881.2160894

[28] V. Stray and 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*, vol. 170, p. 110717, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121220301564

[29] G. Hofstede, *Culture's consequences: Comparing values, behaviors, institutions and organizations across nations*. Sage publications, 2001.

[30] K. H. Roberts and 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.
[31] G. Ailon, "Mirror, mirror on the wall: Culture's consequences in a value test of its own design," *Academy of management review*, vol. 33, no. 4, pp. 885–904, 2008.
[32] R. F. Baskerville, "Hofstede never studied culture," *Accounting, organizations and society*, vol. 28, no. 1, pp. 1–14, 2003.
[33] P. Brewer and S. Venaik, "On the misuse of national culture dimensions," *International Marketing Review*, 2012.
[34] ——, "The ecological fallacy in national culture research," *Organization Studies*, vol. 35, no. 7, pp. 1063–1086, 2014.
[35] A. Sorge, "Review of culture's consequences: International differences in work-related values," *Administrative Science Quarterly*, vol. 28, no. 4, pp. 625–629, 1983. [Online]. Available: http://www.jstor.org/stable/2393017
[36] R. T. Venkateswaran and A. K. Ojha, "Abandon hofstede-based research? not yet! a perspective from the philosophy of the social sciences," *Asia Pacific Business Review*, vol. 25, no. 3, pp. 413–434, 2019.
[37] G. Hofstede, "50 years memory lane – developing cultural dimensions from ibm data," *Software of the Mind 2.0*, Sep 2017.
[38] C. Sadowski and T. Zimmermann, *Rethinking productivity in software engineering*. Springer Nature, 2019.
[39] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 3, pp. 309–346, 2002.
[40] A. Hernández-López, R. Colomo-Palacios, and Á. García-Crespo, "Software engineering job productivity—a systematic review," *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, no. 03, pp. 387–406, 2013.
[41] E. Oliveira, E. Fernandes, I. Steinmacher, M. Cristo, T. Conte, and A. Garcia, "Code and commit metrics of developer productivity: a study on team leaders perceptions," *Empirical Software Engineering*, vol. 25, no. 4, pp. 2519–2549, 2020.
[42] C. Wohlin, M. Höst, and K. Henningsson, "Empirical research methods in software engineering," in *Empirical methods and studies in software engineering*. Springer, 2003, pp. 7–23.
[43] G. Catolino, F. Palomba, D. A. Tamburri, A. Serebrenik, and 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.
[44] B. Vasilescu, A. Serebrenik, and 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.
[45] D. A. Tamburri, F. Palomba, A. Serebrenik, and A. Zaidman, "Discovering community patterns in open-source: a systematic approach and its evaluation," *Empirical Software Engineering*, vol. 24, no. 3, pp. 1369–1417, 2019.
[46] G. Catolino, F. Palomba, D. A. Tamburri, and 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.
[47] F. Palomba and D. A. Tamburri, "Predicting the emergence of community smells using socio-technical metrics: a machine-learning approach," *Journal of Systems and Software*, vol. 171, p. 110847, 2021.
[48] S. MacCurtain, P. C. Flood, N. Ramamoorthy, M. A. West, and 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*, vol. 19, no. 3, pp. 219–232, 2010.
[49] P. M. Blau, *Inequality and heterogeneity: A primitive theory of social structure*. Free Press New York, 1977, vol. 7.
[50] G. Valetto, M. Helander, K. Ehrlich, S. Chulani, M. Wegman, and 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.
[51] L. Williams and R. R. Kessler, *Pair programming illuminated*. Addison-Wesley Professional, 2003.
[52] G. Avelino, L. Passos, A. Hora, and 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.
[53] M. Ferreira, M. T. Valente, and 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.
[54] G. Avelino, E. Constantinou, M. T. Valente, and 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. [Online]. Available: https://doi.org/10.1109/ESEM.2019.8870181
[55] J.-P. Hatala and J. George Lutta, "Managing information sharing within an organizational setting: A social network perspective," *Performance Improvement Quarterly*, vol. 21, no. 4, pp. 5–33, 2009.
[56] M. J. Lindstrom and D. M. Bates, "Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data," *Journal of the American Statistical Association*, vol. 83, no. 404, pp. 1014–1022, 1988.
[57] D. Bates, M. Mächler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *arXiv preprint arXiv:1406.5823*, 2014.
[58] R. M. O'brien, "A caution regarding rules of thumb for variance inflation factors," *Quality & quantity*, vol. 41, no. 5, pp. 673–690, 2007.
[59] A. Cuevas, M. Febrero, and R. Fraiman, "An anova test for functional data," *Computational statistics & data analysis*, vol. 47, no. 1, pp. 111–122, 2004.
[60] K. P. Burnham and D. R. Anderson, "Multimodel inference: understanding aic and bic in model selection," *Sociological methods & research*, vol. 33, no. 2, pp. 261–304, 2004.
[61] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected papers of hirotugu akaike*. Springer, 1998, pp. 199–213.
[62] K. P. Burnham and D. R. Anderson, "Multimodel inference: understanding aic and bic in model selection," *Sociological methods & research*, vol. 33, no. 2, pp. 261–304, 2004.
[63] S. R. de Lemos Meira, E. A. Barros, G. S. de Aquino, and 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.
[64] M. Cataldo, P. A. Wagstrom, J. D. Herbsleb, and 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.
[65] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
[66] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.
[67] B. Penzenstadler, "Towards a definition of sustainability in and for software engineering," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 1183–1185.
[68] B. Penzenstadler, A. Raturi, D. Richardson, C. Calero, H. Femmer, and X. Franch, "Systematic mapping study on software engineering for sustainability (se4s)," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 2014, pp. 1–14.