UNIVERSITY OF SALERNO

DEPARTMENT OF COMPUTER SCIENCE

Master Degree in Computer Science

Thesis in Software Engineering and IT Management

Cultural and Geographical Dispersions Impact on Collaboration and Communication of Software Development Teams

Supervisor:

Prof.
Filomena FERRUCCI

Candidate:

Stefano LAMBIASE
Mat. 0522500778

Co-supervisors:

Prof.
Fabio PALOMBA

Dr.
Gemma CATOLINO

ACADEMIC YEAR 2020/2021

I dedicate this thesis

to the people who work

for those who will come later.

Abstract

Nowadays, Software projects' failure is one of the most critical sources of catastrophic events concerning the business and social world. The most common causes for IT failures are related to project management, which is mainly influenced by social and psychological aspects. The human and social aspects mentioned above became more critical in distributed teams, characterized by the emergence of not-classical problems, for example, cultural differences, language barriers, and high geographical dispersion.

Therefore, this research thesis aims to analyze and illustrate how *cultural and geographical dispersions* (i.e., how much a development community is different in terms of its members' cultural attitudes and geographical collocation) influence the emergence of collaboration and communication problems in open-source communities. To represent the problems mentioned above, we use *community smells*, i.e., sub-optimal patterns across the organizational and social structure in a software development community that are precursors of unforeseen socio-technical events and social debt.

In order to achieve the aforementioned objective, both qualitative and quantitative studies have been conducted (e.g., repository mining, focus groups, surveys, and linear regression) with the aim of: (1) reinforcing the results, (2) covering the problems of each methodology, and (3) being as most comprehensive as possible. Results indicate that such cultural and geographical problems influence collaboration and communication activities, leading to the emergence of community smells in open-source communities. The conclusion is that managers have to know more about cultural and social issues to make better and less risky decisions during the software development process.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Motivations and Objectives

Nowadays, software is pervasive in people's life. For that reason, its development and engineering are essential activities able to influence various aspects, first of all, human life. Therefore, it is important to note that up to 69% of software projects fail [1, 2, 3, 4]. Various research papers reported some consequences of such failures—for example, 17% lead to the company breakdown. The *Virtual Case File Project* (for the United States Federal Bureau of Investigation) is the best documented failure that, according to Frieden [5], led to the loss of five years of development and 170 million USD.

For the aforementioned reasons, knowing more about the core factors that influence project's outcome is compulsory to improve the percentage of successful projects. State of the art has identified various technical and social causes related to two specific fields: *Software Engineering* (SE) and *Project Management* (PM) [1, 4, 3, 6]. Among these factors stand out: (1) project costs under-estimation, (2) lousy risk management, (3) missed staff reward for extraordinary, and (4) failure to communicate and act as a team.

To summarize, it is possible to affirm that Software Engineering and

Project Management are two of the most critical factors influencing the project outcome. In these fields, two problems arise from the modern literature:

- **Social aspects** - Even though Project Management is an activity mainly influenced by human and social aspects [7, 8, 9], most of the research in this field concerns technical processes and metrics.
- **Global Software Engineering** - Nowadays, even more projects are "distributed" (i.e., developed by communities whose members are geographically spread worldwide) with a consequential emergence of collaboration and communication issues [10, 11].

This research thesis' main objective combines both problems, intending to analyze and illustrate how *cultural and geographical dispersions* (that is, how much a development community is different in terms of cultural attitudes and geographical collocation of its members) influence the emergence of collaboration and communication problems.

In order to represent the cultural attitudes, we have chosen to use the ***Hofstede's 6-D framework*** [12, 13], a framework composed of six dimensions (that assume values from zero to one hundred) able to model the cultural aspects of a country and its inhabitants. For instance, two of the six dimensions are *Power Distance Index* and *Individualism vs. Collectivism.* The first one expresses the degree to which the less powerful members of a society accept and expect that power is distributed unequally. The second one represents the degree to which people in a society are integrated into groups.

On the other side, we have decided to represent the social problems using ***community smells***, that is, sub-optimal patterns across the organizational and social structure in a software development community that are precursors of alarming and unforeseen socio-technical events [14, 15, 16, 17]. To better help in the understanding of such "smell", we report, as an example, two classes: *Lone Wolf* and *Organizational Silo.* We have a *Lone Wolf* effect when, within a development community, there are unsanctioned or defiant contributors who carry out their work irrespective or regardless of their peers.

On the other side, we have an *Organizational Silo* effect when a development community presents siloed areas that essentially do not communicate with each other, except through one or two of their respective members.

In order to achieve the aforementioned objective, we have conducted both qualitative and quantitative studies. At first, we have performed various statistical analyses to study the geographical distribution of open-source communities' members, specifically on GitHub (a platform for collaborative software development and version control using Git). Moreover, we have conducted a *focus group* [18] with open source developers, characterized by management experience, to understand how communities perceive cultural dispersion's impact on collaboration and communication in the team. As a quantitative study, we have used *repository mining* and *linear regression* to build a model, which relates cultural and geographical dispersions to the numbers of instances of four types of community smells, i.e., *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*.

State of the art has already demonstrated that cultural and physical distance are common causes of problems, specifically in the collaboration between team members [19, 10, 20, 21]. Nevertheless, very little has been done to formalize such problems and identify precisely the cultural characteristics that are more problematic. Our work is placed on this line of thinking, aiming at (1) formalize the aforementioned problems using community smells, (2) identify new sources for the emergence of such smells, and (3) provide mitigation and refactoring strategies to culturally originated problems.

Main Objective.

Analyze and illustrate, through the use of a mixed-method-research strategy, how cultural and geographical dispersions influence the emergence of collaboration and communication problems, represented using community smells, within software communities.

1.2 Results

In this research thesis, we extend state of the art by providing additional insights into the influence of cultural aspects on software team management. More specifically, we deliver the following novel contribution:

1. **Statistical Analysis** - We conduct a statistical analysis on the developers and development communities on GitHub to explore such entities' cultural aspects and differences.

2. **Focus Group** - We conduct a qualitative investigation in which we recruit six open-source development experts (with different roles and backgrounds) and ask them to be part of a *focus group* [18] to gather opinions on the influence of cultural differences on software development.

3. **Statistical Model** - We build a statistical regression model to study the influence of cultural and geographical dispersions on the presence of community smells in software communities to formalize the link between such cultural problems and communication and collaboration activities.

4. **Packaging** - We provide a GitHub repository [22] to make data and scripts publicly available to provide material that others can use to understand our findings further and build upon our work.

The results of our study reveal that the expected relations between the two types of dispersions—cultural and geographical—and the emergence of communication and collaboration problems are valid. Moreover, the correlation between such dispersions and the presence of community smells in development communities seems true for all the typologies of analyzed smells. For example, both the qualitative and the quantitative research show that the presence of both individualistic and collectivistic people in the same team could lead to *Lone Wolf* effect. These results indicate that managers should consider cultural aspects during the planning and initiation phases of the project to make better-informed decisions. Lastly, the results encourage further the study into the empirical relation between cultural aspects and the complex relationships of Software Engineering organizational structures.

1.3 Document Structure

We have proposed all the topics mentioned above deeply in the seven chapters of which this thesis work is composed. Following is a summary of the chapters:

- **Chapter 2: State Of The Art**, that illustrates the currently available works and discoveries of literature in the field of human aspects and global software development;
- **Chapter 3: Research Methodology**, that illustrates the research questions and the research strategies used to achieve the main objective;
- **Chapter 4: Analysis of The Results**, that illustrates the achieved results for each research question and the conclusions regarding the main objective;
- **Chapter 5: Observations and Lessons Learned**, that illustrates observations on the research and the lessons learned for the research community;
- **Chapter 6: Threats to Validity**, that illustrates the threats to the validity of the study and the way we have mitigated them;
- **Chapter 7: Conclusions and Future Work**, that illustrates a summary of the research and future prospects.

CHAPTER 2

STATE OF THE ART

This chapter illustrates the state of the art and the work present in the literature on research aspects interested by our studio.

2.1 Global Software Engineering

With the growth of the software industry and the emergence of information technology as a pervasive field in everyday life, software product development has drastically changed. One of the most significant changes is known as *Global Software Development* (GSD). Such a switch consists that software development is increasingly a multisite, multicultural, globally distributed undertaking [23]—with all the problems that this comports [24].

Such a shift lead software engineers to face new and important challenges leading to the definition of *Global Software Engineering* (GSE) [25, 26, 10], i.e., the application of *Software Engineering* (SE)[1]—particularly in its *Project Management* (PM)[2] fields—to distributed teams. Undoubtedly, such

[1] *Software Engineering* is "an engineering discipline that is concerned with all aspects of software production" [25]

[2] *Project Management* is "the application of knowledge, skills, tools, and techniques to project activities to meet the requirements" [7]

a distributed working approach comes with many benefits, for example: (1) ability to operate on multiple markets; (2) advantages of widely dispersed time zones; (3) extended productive hours; (4) team skills improvement [27, 23].

Nevertheless, the aforementioned benefits go hand in hand with additional challenges, for example, increased complexity and communication overheads [19, 10, 28]. First of all, allocating and dividing work across different sites or sub-communities is difficult. Mahmood et al. [29] conducted a research to identify the factors that influence task allocation in distributed teams. In addition to identifying various factors, they found that managers are not particularly experienced in this context and that this is one of the many causes for the failure of distributed projects.

A second underestimated issue regards the cultural aspects. GSE requires close cooperation of individuals with different cultural backgrounds. Cultures differ on many critical dimensions, such as the need for structure, attitudes toward hierarchy, sense of time, and communication styles [20]. In their work, Deshpande et al. [20] conducted qualitative research on the impact of culture on GSE. With the help of managers and other figures, they found that culture is one of the most important factors to be counted in Project Management. Moreover, they reported some mitigation strategies to cultural originated problems.

As an ulterior work on cultural influence on GSE, Shah et al. [21] proposed a framework to study such an influence. In contrast to other works, they did not use cultural dimensions (e.g., Hofstede's) to represent culture but a cultural model. Their idea on such cultural models stems from a perspective that sees culture and cognition as mutually constitutive. Thus, the cultural models idea argues that people both understand the world in terms of models and enact models in cultural behavior.

Undoubtedly, collaboration and communication are very challenging in GSE [30, 10]. Stray et al. [30] conducted a longitudinal case study using a mixed-methods approach to study collaboration in GSE. Specifically, they investigated meetings and the well-known collaboration tool Slack[3]—a software platform used to enhance collaboration and communication processes in distributed teams. As a first result, they found that low availability of key people, absence of organizational support for unscheduled meetings, and unbalanced activity from team members in meetings and on Slack are barriers for effective coordination across geographically distributed sub-communities. As a second result, they confirmed the usefulness of communication tools (e.g., Slack) in distributed teams for collaboration tasks. They concluded their work by providing a list of best practices for better meetings and more efficient use of collaboration tools.

2.2 Cultural Aspects in Software Engineering

As said in Section 2.1, software development is, even more, a geographically distributed activity [23, 24]. As a result, Global Software Engineering (GSE) was born to address the challenges—mainly socials—derived from such a nature [26, 30]. Of all the various problems, cultural ones arise as one of the most underestimated and interesting, able to lead to the emergence of catastrophic problems [20, 21].

We can define culture as being: *"socially derived, taken for granted assumptions about how to act and think"* [31]. The culture of individuals can be characterized in two different ways: the first is *national culture*, the second is *organizational culture*. *National culture* regards people's youth education and can be described in terms of shared values, beliefs, and philosophies that affect the way a society is organized [32]. *Organizational culture* can be defined as the set of shared values and beliefs which are seen to characterize particular organizations [33]. In the following section, we report the main studies concerned cultural aspects in Software Engineering, as well as some documented ways to characterize such human nature.

[3] `https://slack.com/intl/it-it/`

2.2.1 Research on Cultural Ascpects

Many researchers have studied the importance and impact of culture as a key factor in Software Engineering. Regarding the GSE and Global Software Development (GSD) field, Krishna et al. [34] explored the impact of cultural aspects and derived some best practices for effective management of distributed teams. Of the various practices stand out: (1) encourage compromise as working culture, (2) set up a team in charge of cultural integration, and (3) perform training sessions on the client culture.

On the same line, Olson and Olson [35] conducted a study on the impact of culture on GSD. As a first contribution, they studied the impact of cultural differences on team composition and teamwork activities, finding various issues that can arise in such contexts. Secondly, they studied in which way various technologies may impact cultural differences in development teams.

Various research has been conducted in the context of *virtual teams*, i.e., groups of developers who work together from different geographic locations and rely on communication technology tools [36, 37, 38]. Rutkowski et al. [37] focused on demonstrating that the use of such tools for communicating—known as *e-collaboration*—could enhance the presence of cultural and stereotypical problems. They also provided a series of best practices to perform a better and more effective e-collaboration.

Interesting work is the one of Calikli et al. [39] on the influence of organizational culture, education, and experience, on *confirmation bias* (namely, the tendency of people to verify their hypotheses rather than refuting them) levels of software developers and testers.

Lastly, Casey[40] wrote a paper to underline the importance of imparting to students the importance of culture to GSD. In his work, he provided a general overview of the concept of culture and the different ways of representing it. Moreover, he evaluated the effectiveness of GSD teaching in providing students with the proper knowledge of cultural concepts.

2.2.2 Hofstede's 6-D Framework

Undoubtedly, culture is a complex topic, hard to be formalized. Nevertheless, tools to represent it are needed in order to measure, compare, evaluate, and predict cultural behavior [41]. Specifically, in the context of GSD, various "dimensions" have been proposed to address such a challenge [12, 42, 43]. In this section, we describe one of the most known frameworks for such an aim: The *Hofstede's 6-D framework* [12].

Geert Hofstede's derived his classification of culture from research carried out in the International Business Machines Corporation (IBM) [12, 44]—so, in the context of Software Engineering and GSD. Hofstede defined culture as *"The collective programming of the mind which distinguishes the members of one group or category of people from another"* [12]. The *Hofstede's 6-D framework* focused his cultural representation on people origin country. He defined six dimensions that assume values from 0 to 100 and which combination characterizes a specific country globally.

The six dimensions are outlined as follows:

Power Distance Index (PDI). *Power Distance Index* expresses the degree to which the less powerful members of a society accept and expect that power is distributed unequally. People in societies exhibiting a high level of *Power Distance* accept a hierarchical order in which everybody has a determined place. On the contrary, in societies with low *Power Distance*, people strive to equalize the distribution of power and demand justification for power inequalities [12, 13]. An example of a case in which this dimension can manifest itself is how people interact with managers, colleagues, and subordinates.

Individualism vs. Collectivism (IDV). *Individualism vs. Collectivism* represents the degree to which people in a society are integrated into groups. A high level of such a dimension indicates a preference for Individualism and for a loosely-knit social framework in which individuals are expected to take care of only themselves and their immediate families. Conversely, a low level of such a dimension indicates a preference for Collectivism and for a tightly-knit framework in society in which individuals can expect their relatives or members of a particular group to look after them in exchange for unquestioning loyalty [12, 13].

Masculinity vs. Femininity (MAS). *Masculinity vs. Femininity* represents a contrast between two preferences. The Masculinity side (high level) is defined as "a preference in society for achievement, heroism, assertiveness and material rewards for success." In contrast, the Femininity side (low level) represents "a preference for cooperation, modesty, caring for the weak and quality of life" [12, 13]. In such a dimension, importance is placed on relationships, helping others, and the physical environment.

Uncertainty Avoidance (UAI). *Uncertainty Avoidance* expresses the degree to which the members of a society feel uncomfortable with uncertainty and ambiguity. Countries exhibiting high level of UAI maintain rigid codes of belief and behavior and are intolerant of unorthodox behavior and ideas. On the other side, a low level of UAI indicates societies that maintain a more relaxed attitude in which practice counts more than principles [12, 13]. UAI is also correlated to the gender disparity problem and to the risk management processes.

Long Term Orientation vs. Short Term Orientation (LTO). *Long Term Orientation vs. Short Term Orientation* measures how much people are oriented toward a long-term outlook in contrast to a more short-term. A high degree in this index (long-term) indicates that people view adaptation and circumstantial, pragmatic problem-solving as necessary. A lower degree of this index (short-term) indicates that people tend to honor traditions and value steadfastness [12, 13].

Indulgence vs. Restraint (IVR). *Indulgence vs. Restraint* refers to the degree of freedom that societal norms give citizens to fulfill their human desires. A high level (Indulgence) indicates a society that allows relatively free gratification of basic and natural human desires related to enjoying life and having fun. In contrast, a low level (Restraint) indicates a society that controls gratification of needs and regulates it using strict social norms [12, 13].

In information systems research, Hofstede's dimensions are widely utilized [45, 40, 46]. For instance, Borchers et al. [45] wrote a paper on the impact of cultural factors on Software Engineering processes. Specifically, they analyzed three distinct cultures: Japan, India, and United States. To investigate how such teams differ in cultural attitudes, they used the Hofstede's framework to represent cultural characteristics. They discovered that different cultures have different approaches to the Software Engineering process (e.g., Japanese people, characterized by a high level of uncertainty avoidance, seem to slow the decision-making process).

Despite such a wide diffusion, some research works raised serious concerns about this framework [47, 48, 49], even calling for the rejection of its use [50, 51, 52]. One of the most cited critiques is that this model links people's cultural attitudes to their origin country, creating an association of the type "you are born here, so you are in this way." Despite this, Venkateswaran and Ojha, in their systematical analysis of the model and its use [53], contend that Hofstede's framework is the most efficient way to represent the complex world of cultural attitudes.

Culture for Hofstede.

Hofstede defines culture as the collective programming of the mind which distinguishes the members of one group or category of people from another. In his view, cultural attitudes are highly related to the birthplace. To model such attitudes, Hofstede defined six dimensions, each related to a specific cultural aspect, that assume values between zero and one hundred and that are defined for most of the countries on the globe.

2.3 Human Aspects in Software Engineering

Since the beginning, the Software Engineering (SE) research community has mainly focused on technical aspects of the software [54, 55, 56, 57, 58]. Such a research activity culminated with the definition of the *technical debt* metaphor—namely, the additional cost caused by programming practices leading to lousy implementation solutions that decrease source code quality—given by Cunningham [59].

Nevertheless, SE is, by nature, a ”social” activity that involves organizations, developers, and stakeholders responsible for leading to the definition of a product that meets the expected requirements [60]. In fact, as early as 1975, Brooks, in his *The Mythical Man-Month: Essays on Software Engineering* [8], mentioned that the composition of teams, and their management techniques, could influence the success of the projects linked to this development. In the following section, we report the main contributions of modern literature to the problem of social aspects in software development and management.

2.3.1 Social Debt

On the social and human perspective in SE, most of the previous literature focused on *social debt*, i.e., unforeseen project costs connected to a suboptimal development community [61, 62]. For instance, Tamburri et al. [62] illustrated social debt by comparison with technical debt and discussed common real-life scenarios that exhibit “sub-optimal” development communities. Always Tamburri et al. [61] conducted exploratory qualitative research in a large software company with the aim to study the causality around social debt in practice. This last work defined a framework containing common social debt causes and some best practices adopted to pay back some of the accumulated debt. Moreover, both the researches revealed that social debt is strongly correlated with technical debt and that both forces should be reckoned with together during the software management process.

Table 2.1: Overview on typical community smells.

Community Smell	Definition
Organizational Silo	Siloed areas of the development community that do not communicate, except through one or two of their respective members.
Black Cloud	Excessive information overload due to a lack of structured communication or cooperation governance.
Lone Wolf	Defiant contributor who apply changes in the source code without considering the considering the opinions of her peers.
Radio Silence	One member interposes herself into every interaction across sub-communities.
Prima Donna	Repeated condescending behavior, superiority, constant disagreement, uncooperativeness by one or few members.
Priggish Members	Demanding of others pointlessly precise conformity or exaggerated propriety, especially in a self-righteous or irritating manner.
Code Red	This smell identifies an area of code which is so complex, dense, and dependent on 1-2 maintainers who are the only ones that can refactor it.
Unlearning	A new technological or organizational advancement or best practice that becomes infeasible when shared with older members.
Disengagement	Thinking the product is mature enough and sending it to operations even though it might not be ready.
Cognitive Distance	The distance developers perceive on the physical, technical, social, and cultural levels regarding peers with considerable background differences.

2.3.2 Community Smells

As an evolution of the research on social debt, Tamburri et al. [14] defined *community smells*, i.e., a set of socio-technical characteristics (e.g., high formality) and patterns (e.g., recurrent condescending behavior, or rage-quitting), which may lead to the emergence of social debt. Table 2.1 reports typical community smells identified by literature [14, 61, 17].

In the last years, community smells have begun to receive particular attention. One of the motivations resides in the development of tools able to detect such smells using various strategies. Tamburri et al. [63] proposed the tool CodeFace4Smells, an augmented version of CodeFace by Joblin et al. [64], capable of detecting community smells. Moreover, Tamburri et al. [63] assessed the detection capabilities of CodeFace4Smells using a survey and performing an empirical evaluation on a set of 60 projects. Such activity confirmed the capacity of the tool in the detection of community smells. In addition to that, Palomba et al. [15] used CodeFace4Smells to investigate the impact of such smells over *code smells*.[4] Such a study has reported that community smells represent key factors preventing developers from performing refactoring activities.

A more recent and promising trend is represented by the definition of modeling mechanisms able to describe the future structure of a community and alert project managers of the presence of social debt [66, 67, 68]. As a contribute to this, Almarimi et al. [67] build a multi-label learning model based on genetic algorithms to detect eight common types of community smells. The tool evaluation involved 103 open-source projects and 407 community smells instances and resulted better performance indexes compared to other solutions (F-measure of 89%).

Researchers have also investigated the significant factors that influence community smells and the way managers refactor them. Catolino et al. [69] conducted an empirical study investigating the correlation between gender diversity and the emergence of four types of community smells. Such a paper showed how the emergence of community smells might be potentially reduced by the presence of women in development teams. One of the most exciting results consists of women's influence on the emergence of smells correlated to the quality of communication (e.g., *Radio Silence* effect). Specifically for the refactoring strategies, Catolino et al. [70] also studied how developers remove community smells. Through a survey conducted with 76 experts, the authors elicited a set of refactoring operations generally applied by practitioners to remove four types of smells. From the identified strategies emerge mentoring, monitoring, creation of a communication plan, conducting cohesion exercises and restructuring the community.

For which concern the impact that such community smells have on the whole software development field, various works have been conducted. Martini and Bosch [71] studied the impact of community smells on architecture debt,

[4] *code smells* are "poor implementation choices applied by developers during software evolution that often lead to critical flaws or failure" [15] [65].

while Tamburri et al. [72] on organization structure types.

Community smells.

Community smells are a set of socio-technical characteristics (e.g., high formality) and patterns (e.g., recurrent condescending behavior, or rage-quitting) in a software development community, that are precursors of alarming and unforeseen socio-technical events like social debt [14, 17].

2.3.3 Socio-Technical Quality Factors Metrics

Software project development involves two fundamental elements: a technical and a social component. The technical component can be considered as the combination of (1) technical properties of the product to develop, (2) the development processes, and (3) the technologies employed during the product development cycle. The social component consists of (1) the organization and the individuals involved in the development process, (2) their attitudes, (3) their behaviors, and (4) the relationships between them. Undoubtedly, a product development project can be seen as a *socio-technical system* where the two components discussed above need to be aligned and managed to grant a successful outcome. Moreover, it should not be surprising that the two components can influence each other and how managing this relationship requires considerable effort.

In order to represent and operationalize such a socio-technical behavior, state of the art proposed various metrics. Tamburri et al. [16] conducted a *systematic literature review* (SLR) that investigated social and socio-technical factors for software engineering [73, 62, 74, 75]. Table 2.2 reports these factors by dividing them by category and providing a brief definition. Moreover, Magnoni et al. [76] proposed a *Socio-technical Quality Framework*, composed of factors reported in Table 2.2, to study which of them mainly influence the emergence of social debt.

Socio-Technical Congruence

Besides the studies on social debt and socio-technical metrics, several empirical analyses have been carried out on the so-called *Socio-Technical Congruence* [77, 78, 79, 27]. The main idea of *Socio-Technical Congruence* is based on *Conway's Law*, named after Melvin Conway, which said, *"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure."* [80]. Based on this law, *Socio-Technical Congruence* (STC) can be defined as a measure that asses the alignment between coordination requirements (extracted from technical dependencies among tasks) and the actual coordination activities performed by the developers [77, 78].

Various works have demonstrated the benefits of gaining a good level of STC, specifically on productivity [77] and quality [81]. Moreover, some tools that can measure STC and STC-related aspects have been proposed, like Ariadne [82] or Tesseract [83], designed to visualize coordination relationships and gaps in software communities. In the context of STC in Global Software Engineering, Portillo-Rodríguez et al. [27] designed an Agent Architecture for coordination and communication ables to measure STC in distributed teams. Such a proposal has been validated through a case study performed at Indra Software Labs with positive results.

More related to our study, Palomba et al. [15] conducted an empirical investigation to address the co-existence and compounding effects between community and code smells. Always Palomba et al. [84] surveyed various developers to preliminarily assess if such an influence exists. Such studies have reported that community smells represent key factors preventing developers from performing refactoring activities.

Table 2.2: Social and Socio-Technical Quality Factors [16].

Category	Metric	Description
Developer Social Network metrics	devs	Number of developers present in the global Developers Social Network
	ml.only.devs	Number of developers present only in the communication Developers Social Network
	code.only.devs	Number of developers present only in the collaboration Developers Social Network
	ml.code.devs	Number of developers present both in the collaboration and in the communication DSNs
	perc.ml.only.devs	Percentage of developers present only in the communication Developers Social Network
	perc.code.only.devs	Percentage of developers present only in the collaboration Developers Social Network
	perc.ml.code.devs	Percentage of developers present both in the collaboration and in the communication DSNs
	sponsored.devs	Number of sponsored developers (95% of their commits are done in working hours)
	ratio.sponsored	Ratio of sponsored developers with respect to developers present in the collaboration DSN
Socio-Technical metrics	st.congruence	Estimation of socio-technical congruence
	communicability	Estimation of information communicability (decisions diffusion)
	num.tz	Number of timezones involved in the software development
	ratio.smelly.devs	Ratio of developers involved in at least one Community Smell
Core community members metrics	core.global.devs	Number of core developers of the global Developers Social Network
	core.mail.devs	Number of core developers of the communication Developers Social Network
	core.code.devs	Number of core developers of the collaboration Developers Social Network
	sponsored.core.devs	Number of core sponsored developers
	ratio.sponsored.core	Ratio of core sponsored developers with respect to core developers of the collaboration DSN
	global.truck	Ratio of non-core developers of the global Developers Social Network
	mail.truck	Ratio of non-core developers of the communication Developers Social Network
	code.truck	Ratio of non-core developers of the collaboration Developers Social Network
	mail.only.core.devs	Number of core developers present only in the communication DSN
	code.only.core.devs	Number of core developers present only in the collaboration DSN
	ml.code.core.devs	Number of core developers present both in the communication and in the collaboration DSNs
	ratio.mail.only.core	Ratio of core developers present only in the communication DSN
	ratio.code.only.core	Ratio of core developers present only in the collaboration DSN
	ratio.ml.code.core	Ratio of core developers present both in the communication and in the collaboration DSNs
Turnover	global.turnover	Global developers turnover with respect to the previous temporal window
	code.turnover	Collaboration developers turnover with respect to the previous temporal window
	core.global.turnover	Core global developers turnover with respect to the previous temporal window
	core.mail.turnover	Core communication developers turnover with respect to the previous temporal window
	core.code.turnover	Core collaboration developers turnover with respect to the previous temporal window
	ratio.smelly.quitters	Ratio of developers previously involved in any Community Smell that left the community
Social Network Analysis metrics	closeness.centr	SNA degree metric of the global DSN computed using closeness
	betweenness.centr	SNA degree metric of the global DSN computed using betweenness
	degree.centr	SNA degree metric of the global DSN computed using degree
	global.mod	SNA modularity metric of the global DSN
	mail.mod	SNA modularity metric of the communication Developers Social Network
	code.mod	SNA modularity metric of the collaboration Developers Social Network
	density	SNA density metric of the global Developers Social Network

CHAPTER 3

RESEARCH METHODOLOGY

This chapter illustrates the research questions and the research strategies used to achieve the main objective.

The study's primary goal is to assess to what extent cultural and geographical dispersions—how much a development community is different in terms of its members' cultural attitudes and geographical collocation—in software communities influence such communities' communication and collaboration activities. We want to provide a deeper understanding of the way globally distributed software teams communicate and cooperate. The perspective is that of project managers, who are interested in effectively allocating resources or managing complex organizational structures.

3.1 Hypothesis and Research Questions

The working hypothesis behind this work is stated as follows:

The presence inside development communities of people from different cultures can impact the collaboration and communication activities, leading to the emergence or mitigation of community smells.

In order to facilitate the objective achievement and to better study the previously described hypothesis, we have separated the research process into five steps and, consequentially, into the following research questions:

RQ_1 *Which are the cultural and geographical characteristics of open-source communities and developers on* GitHub*?*

RQ_2 *How much are the development communities on* GitHub *culturally dispersed?*

RQ_3 *In which way do cultural characteristics of development communities change over time?*

RQ_4 *In which way do cultural and geographical dispersions impact collaboration and communication activities in open-source communities?*

RQ_5 *To what extent do cultural and geographical dispersions within teams influence the number of community smells?*

The first research question intends to be a preliminary analysis. It aims to provide an overview of the culture and geography characteristics of GitHub developers using various statistics, that is, mean and standard deviation. As the first one, the second research question aims to provide an overview but at the level of development communities. The objective is to study how much the development communities are composed of developers from different cultures and how much they are culturally different. With the third research question, we want to study the evolution of communities' cultural characteristics during the project's progress. Instead, the fourth research question aims to collect and analyze developers' impressions regarding the influence of cultural and geographical differences inside a community on collaboration and communication problems. The last research question arises from the necessity of formalizing the previously discussed social problems using community smells—that is, patterns of sub-optimal organisational and socio-technical characteristics that may lead to tangible problems in

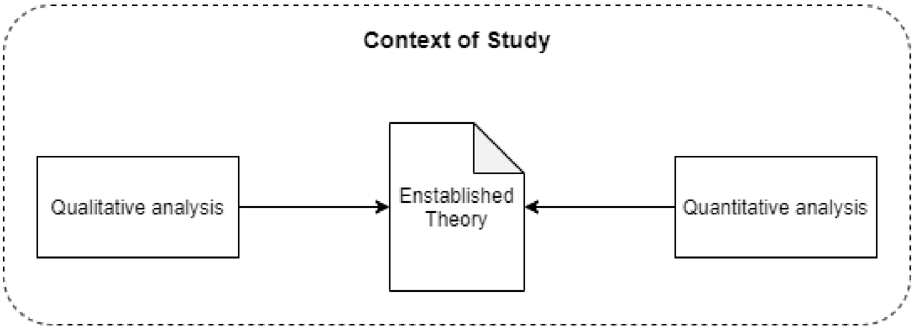

Figure 3.1: Mixed-Methods approach general model.

development communities [14]. In that phase, through repository mining and linear regression, the cultural and geographical dispersions are correlated with a subset of community smells that better represent collaboration and communication problems. Such a subset of smells is composed by: *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*.

In order to represent the cultural aspects, we have chosen to use the *Hofstede 6-D model* [12, 13], a framework composed of six dimensions able to model the cultural aspects of a country and its inhabitants.

3.2 Mixed-Methods Research Approach

With the aim of being more comprehensive and strengthening research results, we adopted a *mixed-methods* approach (also known as *mixed-methods research*). As shown in Figure 3.1, quantitative and qualitative research processes are run in parallel over the same dataset in these types of strategies [85, 86, 87]. The ultimate goal of such mixed-methods is converging towards theoretical saturation (i.e., "the point in category development at which no new properties, dimensions, or relationships emerge during analysis" [88]). Therefore, we assume that if both qualitative and quantitative data lead to the same conclusions, then our theory is saturated. Conversely, any disagreement between the two theories would lead us to an improved version of our initial theory that cultural characteristics and social problems are somehow connected. For the quantitative part, we have used, mainly, *repository mining*

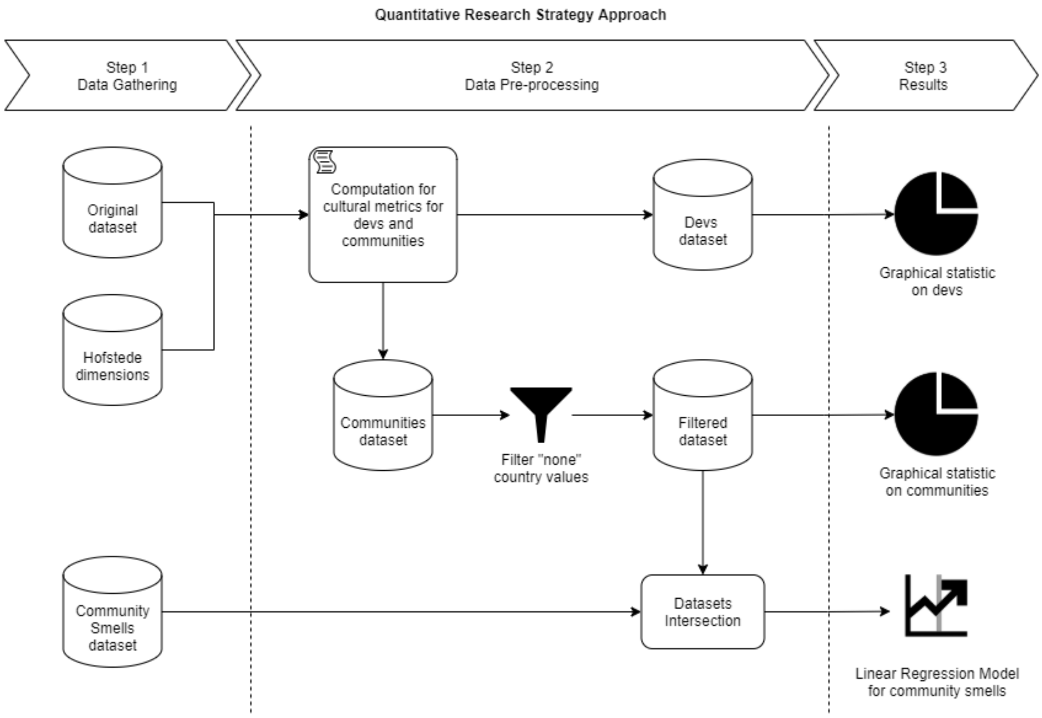

Figure 3.2: Overview of our quantitative methodology.

and *linear regression model construction*. For the qualitative part, we have performed a *focus group* with six developers.

3.3 Context of Study

The context of the study consisted of software communities, community smells, and open-source developers.

In Figure 3.2, we summarized the research process for our quantitative studies. For the first three research questions, we needed to compute cultural and geographical information of the developers and communities on GitHub. Thus, we started from the dataset made available by Vasilescu, Serebrenik, and Filkov [89]. The dataset reports information on 23 493 projects and corresponding communities and developers. Therefore, we used the information about the origin country of each developer to compute the Hofstede cultural dimensions for each of the communities' members and each development community. In the second case, we excluded teams with a high

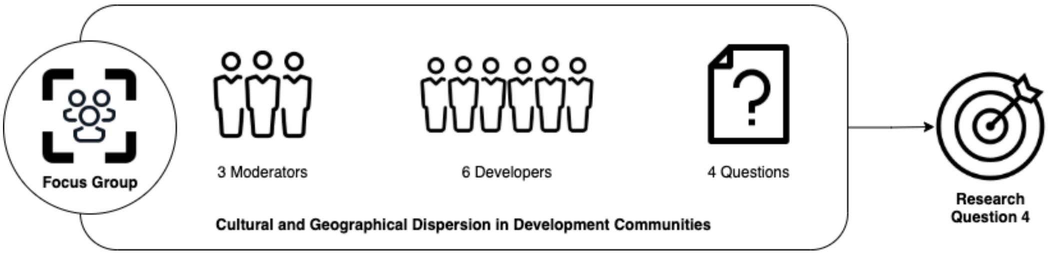

Figure 3.3: Overview of our qualitative methodology.

level of missed values for the members' origin country, intending to take only the better-characterized ones.

For the fourth research question, we needed to qualitatively study the influence of cultural and geographical dispersions on communication and collaboration activities. Figure 3.3 summarized the research process for our qualitative methodology. We recruited six open-source developers with long work experience in distributed teams, also cross-continental. They all work in the industry and the context of open-source development, specifically, from a company that is known for being active in open-source solutions development and promoting distributed and social-oriented processes, as well as conducting research in human integration in software development. While all the participants have experienced working in a distributed team in the same continent, two of them also participated in development communities cross-continental. We chose six participants to fulfill the research approach employed to address the fourth research question: as Wilkinson said in his work [18], *focus groups* are a qualitative research strategy that involves a small number of people capable of providing expert judgments on the subject of interest. Specifically, we chose six participants following established guidelines that state that the ideal size of a focus group is five to eight people [90].

For the last research question, we needed to study the correlation between the two types of dispersions—cultural and geographical—and the number of instances of four community smells, i.e., *Organizational Silo*, *Black Cloud*, *Lone Wolf*, and *Radio Silence*. The four smells are outlined as follows:

Organizational Silo Effect. We have an *Organizational Silo* effect when a development community presents siloed areas (or sub-communities) that essentially do not communicate with each other, except through one or two of their respective members. The *Organizational Silo* reflects rigid thinking, narrow tunnel-vision, as well as lack of community-wide communication.

Black Cloud Effect. We have a *Black Cloud* effect when a development community exhibits an excessive information overload due to a lack of structured communication or cooperation governance. The *Black Cloud* reflects the presence of overwhelming quantities of async and synch data exchanges across a development community.

Lone Wolf Effect. We have a *Lone Wolf* effect when, within a development community, there are unsanctioned or defiant contributors who carry out their work irrespective or regardless of their peers. *Lone Wolves* often manifest concomitant with components or software modules that were explicitly modularized for cooperation (contemporary operation over software artifacts) instead of collaboration between peers (joint work on shared artifacts) [91].

Radio Silence Effect. We have a *Radio Silence* effect when, within a development community, one member interposes herself into every formal interaction across two or more sub-communities with little or no flexibility to introduce other parallel channels. Specifically, *Radio Silence* is an instance of the "unique boundary spanner" [92] problem from social-networks analysis.

It is worth noting that we can group the considered smells into two high-level categories: structural-based, for *Organizational Silo* and *Black Cloud*, and communication-based, for *Lone Wolf* and *Radio Silence*. Therefore, as shown in Figure 3.4, those smells allow us to formalize the aforementioned communication and collaboration problems in order to combine qualitative and quantitative research processes. We used the previously computed cultural

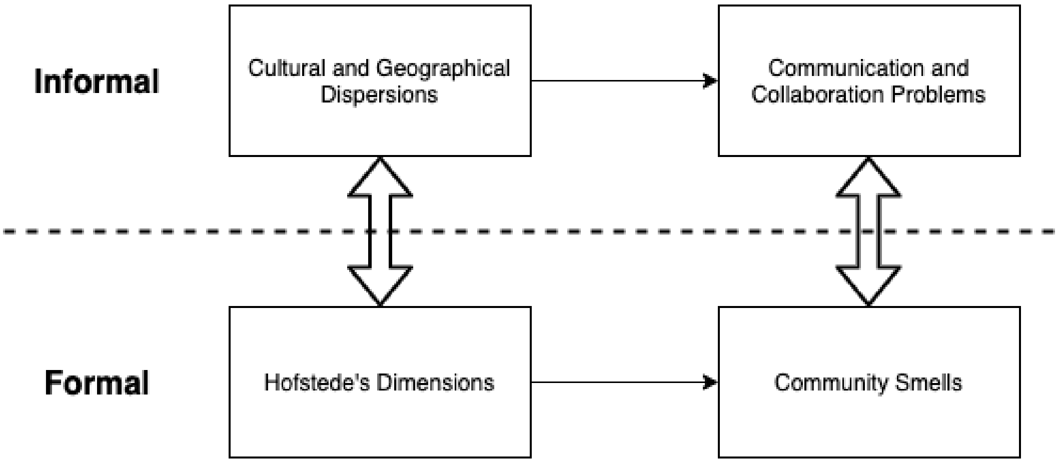

Figure 3.4: Research components formalization.

and geographical information only for the communities present in both the Vasilescu et al. [89] and the Catolino et al. [69] datasets. Those communities correspond to 20 projects for which we had the number of community smells for each of the previously cited four categories.

3.4 $\mathbf{RQ_1}$ - Culture and Geography of GitHub

> **$\mathbf{RQ_1}$**. Which are the cultural and geographical characteristics of open-source communities and developers on GitHub?

To address **$\mathbf{RQ_1}$**, we conducted descriptive statistics strategies on the cultural and geographical characteristics of open-source communities and their members on GitHub. We collected data from the dataset used by Vasilescu et al. [89] that reports information on the open-source teams, and the corresponding members, of 23 493 projects. To represent the cultural aspects, we have calculated the six dimensions of Hofstede for each developer.

The developers' origin country information was already present in the used datasets, retrieved using the GitHub APIs[1] and the developers' unique identifiers. It is important to note that the location descriptions on profile pages are free-text optional entries. For that reason, they

[1] `https://docs.github.com/en/rest`

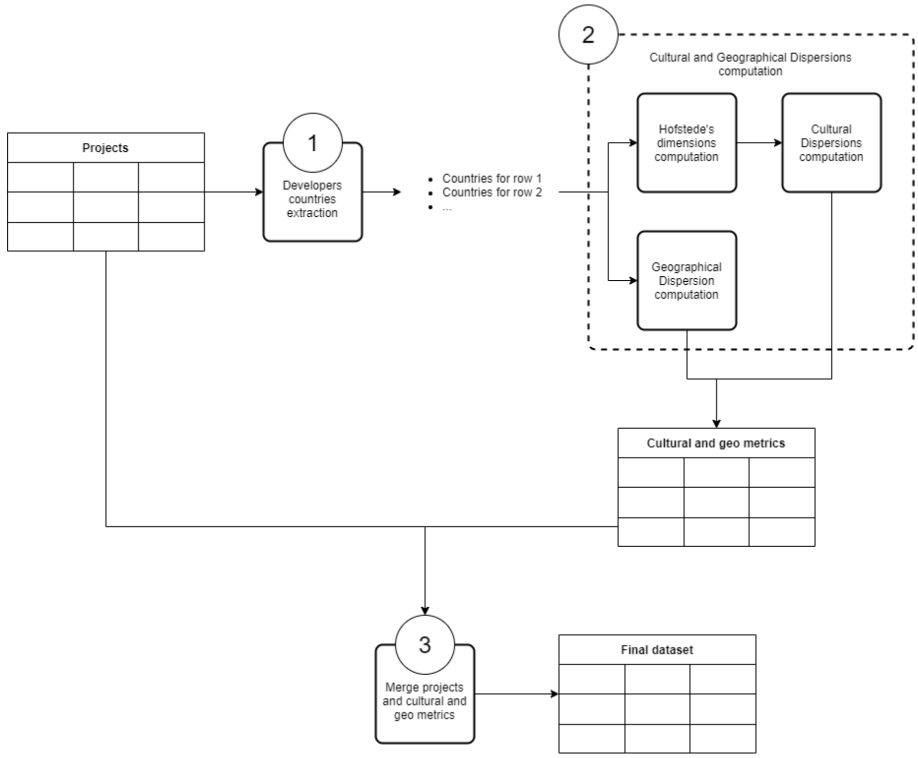

Figure 3.5: Overview of cultural and geographical metrics computation.

include unstructured and often noisy data, for example, latitudes/longitudes, postcodes, IP-addresses, and fictitious addresses (e.g., "the most distant place from the center of the universe"). Therefore, after data preprocessing, only a tiny part of the developers in the dataset have associated a "credible" country, that is, 339 102 (86.7%) of the users with non-empty location entries.

Figure 3.5 describes the process to calculate the cultural and geographical metrics. From the Vasilescu et al. dataset [89], we extracted, for each development community, all members' IDs and associated countries. After that, we constructed two datasets: the first one containing the number of developers on GitHub for each country, the second one containing the important information of each developer (e.g., ID, gender, and origin country). We used the first dataset to represent the geographical structure of GitHub and the second one to compute Hofstede's dimensions values for each developer:

- **pdi**, that contains values for the *Power Distance Index* dimension;
- **idv**, that contains values for the *Individualism vs Collectivism* dimension;

- **mas**, that contains values for the *Masculinity vs Femininity* dimension;
- **uai**, that contains values for the *Uncertainty Avoidance* dimension;
- **lto**, that contains values for the *Long Term Orientation vs Short Term Orientation* dimension;
- **ivr**, that contains values for the *Indulgence vs Restraint* dimension.

We used the second dataset also to calculate the average and the standard deviation of the developers' values for each Hofstede's measure. We applied the standard deviation as a measure to represent how much the developers of GitHub have different ideas on cultural attitudes.

In the end, we have used *pie plots* and *bar plots* to represents the geographical information of developers. We have used *violin plots* to represents Hofstede's dimensions values associated with the number of developers.

3.5 $\mathbf{RQ_2}$ - Cultural Dispersion of GitHub Communities

$\mathbf{RQ_2}$. How much are the development communities on GitHub culturally dispersed?

The *cultural dispersion* of a development community is the degree in which developers form a community with different cultural attitudes. A low level indicates that the community members come from the same cultural education; the opposite, a high level indicates that the community members present different cultural attitudes. In our case, to operationalize the cultural dispersion of a community we have used the Hofstede's 6-D model. Specifically, we have chosen to represent cultural dispersion using six values, one for each Hofstede dimension. Each of these values corresponds to the standard deviation—the cultural dispersion—of the set containing the community members' values for one of the six dimensions.

Starting from the Vasilescu et al. dataset [89], we computed the values, average, and standard deviation for each Hofstede dimension per development community. Then, we created a dataset combining the one from Vasilescu and the cultural measures mentioned above. Consequentially, we used graphical representations to show (1) the cultural average values of development communities and (2) how much they are culturally dispersed. We chose to represent only the values corresponding to the communities for which we knew more than 50% of the members' countries (500 projects, with multiple time window). We intended to take only the communities for which the computed cultural characteristics were the most accurate.

Cultural dispersion.

We defined *cultural dispersion* as the degree in which developers form a community with different cultural attitudes. A low level indicates that the community members come from the same cultural education; the opposite, a high level indicates that they present different cultural attitudes.

3.6 RQ$_3$ - Cultural Variation in Time

RQ$_3$. In which way do cultural characteristics of development communities change over time?

In order to study the evolution of the cultural dispersion of development communities, we have formulated the **RQ$_3$**. We aimed to verify if, with the continuation of the development project, the communities tend to attract people from different or same cultures of the developers already in the team. To achieve this, we took advantage of the Vasilescu et al. [89] dataset that provides different temporal windows for the same project—and the corresponding community—each one separated by 90 days. We considered only communities for which we knew more than 50% of the members' countries (for reliability in the computation of cultural characteristics) and at least three temporal windows (to rule out projects that have not evolved much). After we divided all the temporal instants for the various communities, for each one, we performed the following steps:

1. we calculated the increment (or decrement) of each one of the Hofstede measures from one temporal window to the successive;

2. we calculated the average and the standard deviation of the list of increments (and decrements): the first represents the *cultural variation* of the community in time, and the second represents if the variations of the dimension of all the windows pairs are similar or not.

In the end, we use *violin plots* to represents the two obtained data (average and standard deviation) for all the communities under consideration.

Cultural variation.

We define *cultural variation* as the rank in which the culture of a community tends to change over time. A value near zero indicates that the community does not experiment with any change in its culture; a value different than zero indicates that the culture of the community changes.

3.7 $\mathbf{RQ_4}$ - Focus Group with Open-Source Developers

$\mathbf{RQ_4}$. In which way do cultural and geographical dispersions impact collaboration and communication activities in open-source communities?

Recalling our aim to perform a mixed-method-research approach, we formulated **$\mathbf{RQ_4}$** and chose to address it using a qualitative approach, namely, a focus group. Previous works in the Software Engineering field have exploited *survey* research as an efficient way to complement quantitative studies, e.g., mining [93]. In Software Engineering, a *survey* is a research method used for collecting data from a predefined group of respondents to gain information and insights on an event that included such participants [94]. Even though surveys provide quantifiable data usable to establish, on a large scale, a theory, they are not interactive and are likely to lack details or depth on the investigated topic [95]. As the goal of $\mathbf{RQ_4}$ was to gather opinions on the influence of cultural and geographical differences on collaboration and communication, we then preferred to perform a more qualitative approach like focus groups, characterized by having the advantage of fostering discussions and uncovering ideas that otherwise would have been missed [18, 95]. We did not exclude entirely from using a survey to strengthen and generalize the results obtained from the focus group. Indeed, as Fenton and Bieman said in their work [94], a survey is a retrospective study of a situation to try to document relationships and outcomes; thus, it is done after an event has occurred.

In the context of our study, we performed a focus group to enable a joint discussion on the cultural and geographical differences of the developers inside development communities. We also discussed the possible problems that arise from such differences and some strategies to mitigate them. From a practical standpoint, we invited the participants to join an online Zoom meeting. A physical meeting was not possible because participants come from different countries and because of the COVID19 pandemic.

We organized the focus group into two main parts: the first one acted like an introduction on cultural dispersion and ended with the asked questions; the second one consisted of the practical discussion on the topic by all participants.

The first author acted as moderator and presenter. At the beginning of the meeting, after a brief presentation, the moderator shared some slides to help the group focus on the topic. The presentation provided to the participants is available in our online appendix [22]. More specifically, the aim was to provide an overview of the cultural dispersion concept, show some examples of cultural attitudes and differences according to the Hofstede framework, and introduce the four questions asked:

1. *"Did you identify different cultural attitudes from the previous examples?"*

2. *"There is a project or a particular experience in which you have seen different people with different cultures collaborate?"*

3. *"These differences impacted the collaboration and communication?"*

4. *"If there were problems, which strategies did you have used (or you have seen used) to mitigate and refactor them?"*

The first question aims to understand if the developers agree that the Hofstede cultural measures are capable of characterizing the cultural differences in software development teams. It is a way to validate the effectiveness of the Hofstede framework in the Software Engineering field and identify different cultural attitudes from the Hofstede ones. The second question aims to help participants recover memories of past experiences to allow them to discuss the topic better and break the ice. The third question aims to identify the previously discussed collaboration and communication problems—that are the core aspects of this thesis work—and link them to the cultural differences. The last question aims to identify some possible mitigation strategies of such communication and collaboration problems that originated from cultural and geographical differences. More precisely, we want to study how people react to these problems and if there are some patterns in facing them.

In the second part of the focus group, participants were asked to respond to the four aforementioned questions and informally discuss the cultural dispersion topic concerning their past experiences. This part of the meeting lasted 50 minutes and was kept by the moderator highly interactive. The moderator did not simply leave the word to each participant but asked others to comment and reflect on the possible reasons behind what s/he was reporting. The entire discussion was recorded and stored for analysis.

Upon completion, ZOOM provided as output the video registration of the meeting. This registration was reviewed in order to identify the main insights and comments left by the participants. We finally addressed $\mathbf{RQ_4}$ by reporting the most relevant insights from the focus group.

Table 3.1: Projects with community smells information.

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

3.8 RQ$_5$ - Building a Statistical Model

RQ$_5$. To what extent do cultural and geographical dispersions within teams influence the number of community smells?

To address **RQ$_5$**, we defined a statistical model relating a development community's cultural and geographical dispersions to the frequency of community smells in the same group. We created the used dataset combining the ones used by Vasilescu, Serebrenik, and Filkov [89] and Catolino et al. [69]. The first dataset reports information on the open-source teams and the corresponding members of 23 493 projects, divided into time bands of 90-days. For some of the first dataset projects, the second dataset contains information regarding the number of instances of four community smells, i.e., *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*. Thus, we took the analyzed projects, and the frequency of community smells by the Catolino et al. [69] dataset. At the same time, we calculated the cultural and geographical metrics on the Vasilescu, Serebrenik, and Filkov [89] dataset (only for the projects also present in the Catolino et al. dataset) using the country of each developer, already given in the data. Table 3.1 reports the projects contained in the second dataset with the associated number of temporal windows and the programming language.

As already said in Section 3.4, in the first dataset, the developer's country has been inferred through the GitHub APIs[2], using the developers' unique identifier to retrieve the country inserted by the user in his account. It is important to note that the location descriptions on profile pages are free-text optional entries. For that reason, they include unstructured and often noisy data. Therefore, after data preprocessing, only a tiny part of the developers in the dataset have associated a "credible" country, that is, 339 102 (86.7%) of the users with non-empty location entries.

3.8.1 Independent Variables

Based on our hypothesis, we considered seven factors as independent variables. The first six represent the cultural dispersion of a development community: the rank in which developers form a community with different cultural attitudes (measured using the Hofstede 6-D model). As in Section 3.5, to operationalize the cultural dispersion of a community, we have chosen to use six values, one for each Hofstede dimension. Each of these values corresponds to the standard deviation of the set containing the community members' values for one of the six dimensions. In this way, we obtained the following six factors:

- **Power Distance Index Dispersion (PDID)** - *PDID* indicates how much community members tend to have a different idea on hierarchical

[2] `https://docs.github.com/en/rest`

structure and power division.

- **Individualism vs. Collectivism Dispersion (IDVD)** - *IDVD* indicates how much community members tend to have a different idea regarding forming groups and sharing the success.
- **Masculinity vs. Femininity Dispersion (MASD)** - *MASD* indicates how much community members tend to have a different opinion about self-affirmation and help the weaker elements.
- **Uncertainty Avoidance Dispersion (UAID)** - *UAID* indicates how much community members tend to have a different idea on taking risks and accepting new and controversial opinions.
- **Long Term Orientation vs. Short Term Orientation (LTOD)** - *LTOD* indicates how much community members tend to have a different opinion about investing in the future and conserving old traditions.
- **Indulgence vs. Restraint Dispersion (IVRD)** - *IVRD* indicates how much community members tend to have a different opinion about the rank in which the authority has to control satisfying people needs.

The last factor regard the geographical distance between members:

- **Geographical Dispersion** - The *Geographical Dispersion* of a development community is computed as the standard deviation of the set of physical distances between each community member—calculated using the origin country already present in the dataset.

3.8.2 Response Variables

We aimed at understanding the impact of cultural and geographical dispersions on the presence of community smells. For that reason, our response variable was represented by the number of community smells present in the Catolino et al. dataset. This information has been calculated using the tool

CodeFace4Smells (namely, an automated approach able to identify four community smell types, that is, *Radio Silence*, *Lone Wolf*, *Black Cloud*, and *Organizational Silo*) by Tamburri, Palomba, and Kazman [63].

Response variable: instances of Community Smells.

The response variable is the number of instances of four types of community smells, i.e., *Organizational Silo*, *Lone Wolf*, *Radio Silence*, and *Black Cloud*, within a development community.

3.8.3 Control Variables

While our hypothesis pertained to the relationship between the two types of dispersions—cultural and geographical—on the presence of community smells, it is vital to note that many other community-related factors might influence the response variable. To account for this, we consider various control variables. From the first dataset [89], we consider the following variables:

- **Number of Committers** - We define *Number of Committers* as the number of people that have done at least one commit in a given project time window. Catolino et al. [69] have demonstrated that the number of committers in a community can influence the number of community smells. Therefore, we considered the total committer count of a project as the first control variable.

- **Number of Commits** - We define *Number of Commits* as the total commit count in a project. In the majority of the case, a high number of commits may indicate a high activity in the community. Causing that it is not rare that more developers work on the same code module or functional requirement, such an activity might be corresponding with high communication and collaboration, possibly impacting the number of community smells.

- **Team Size** - We define *Team Size* as the number of contributors per project team in a given temporal window. The community's size can influence the number of community smells it contains, for example, incrementing the number of smells with an increment of community members.

- **Turnover** - We intend *Turnover* as the fraction of the team in a given temporal slice that is different with respect to previous windows (i.e., the *turnover ratio*). A high turnover indicates that community members changed frequently. Vasilescu, Serebrenik, and Filkov [89] have demonstrated that the constant introduction of new contributors might lead to communication and collaboration problems.

- **Project Age** - We define *Project Age* as the difference between the maximum index and the index of the 90-day temporal interval during which the first commit was recorded. Latter projects and their teams may have experienced different trends or work habits, and these changes might affect the presence of community smells.

- **Tenure diversity** - *Tenure measures* have been used in state of the art to represents the experience of developers in various fields [96, 97], showing how are able to influence communication and collaboration tasks. In our dataset, we considered two typologies of tenures: (1) *commit tenure* (that measures the coding experience of a contributor within all GitHub projects in which he has contributed) and (2) *project tenure* (that measures the developer experience in the specific project considered). As tenure measures are numerical variables, the dataset report them using the coefficient of variation, that is, the ratio between the adjusted standard deviation and the mean.

- **Tenure median** - Remaining in the field of tenure metrics, to represent an average of project and commit tenures for what concern developers of a community, also the project and commit median tenure has been considered.

- **Number of women in a team** - The number of women is computed as the difference between the total number of community members and the number of men belonging to the community. Catolino et al. [69] demonstrate that number of women can influence the occurrences of some community smells, that is, *Black Cloud* and *Radio Silence*.

- **Blau-Index** - Blau [98] defined *Blau diversity index* as: $BLAU = 1 - \sum_{i=1}^{n} P_i^2$ 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.

From the second dataset [69], we consider the following variables:

- **Socio-Technical Congruence** - As defined by Valetto et al. [99], *Socio-Technical Congruence* represents "the state in which a software development organization harbors sufficient coordination capabilities to meet the coordination demands of the technical products under development" [99]. Catolino et al. [69] have operationalized it as the number of development collaborations that do communicate over the total number of collaboration links present in the collaboration network.

- **Truck Factor (TF)** - We defined *Truck Factor (TF)* (also known as *Bus Factor*) as the minimum number of member of a community that have to quit (or being hit by a bus) before the project will fail [100, 73, 101]. In their dataset, Catolino et al. [69] operationalized truck factor based on core and peripheral community structures identified by CodeFace4Smells [63], as the degree of ability of the development community to remain connected without its core part.

- **Centrality** - We define *Centrality* as the strength of a community and it is based on modularity measures [102]. A value over 0.3 indicates that the community is highly modular and thus with a clear distinction of the sub-communities present in its development network. On the other side, a value below 0.3 indicates that there are no sub-communities.

3.8.4 Statistical Model Construction

The response variables consist of the number of instances of four types of community smells: *Organizational Silo*, *Black Cloud*, *Lone Wolf*, and *Radio Silence*. For that reason, we built four statistical models, i.e., one for each community smell considered, all with the same independent and control variables. Since the dataset was composed of multiple temporal windows for each project, we had to build four *linear mixed models*[3] in order to capture measurements from within the same group (i.e., within the same project) as a random effect [103]. In our case, we used the time window as a random effect and all other variables as fixed effects.

To strengthen the results, we use two different technologies to build the model: the Python library `statsmodel` [104] and the `R` package `lme4`.[4]

Using the `statsmodel` PYTHON library, we built four multiple linear mixed effects models thanks to the function `mixedlm`[5]. The function `mixedlm` handles most non-crossed random effects models, and some crossed models and uses the *two-tailed* test [105] to compute the statistical significance of the resulted parameters. Using the functions `lmer` and `lmer.test` available in the `R` package `lme4` [106], we rebuilt the same models. In order to avoid the problem of multi-collinearity [107], we employed a stepwise variable removal procedure based on the *Companion Applied Regression* (`car`) `R` package,[6] and in particular based on the `vif` function [107]. Finally, the effect sizes of the coefficients were obtained using the ANOVA statistical test [108] and were considered important if they were statistically significant (that is, the $\rho - value$ is less than 0.05).

To run the computation and show the results we used two typologies of

[3] *Linear mixed models* are an extension of simple linear models that take into account the data's hierarchical structure. They allow specifying two sets of variables: (1) fixed factors, which are the independent variables of interest to the study, and (2) random factors, which are the classification variables that represent the data groups.

[4] https://cran.r-project.org/web/packages/lme4/lme4.pdf

[5] https://www.statsmodels.org/stable/mixed_linear.html

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

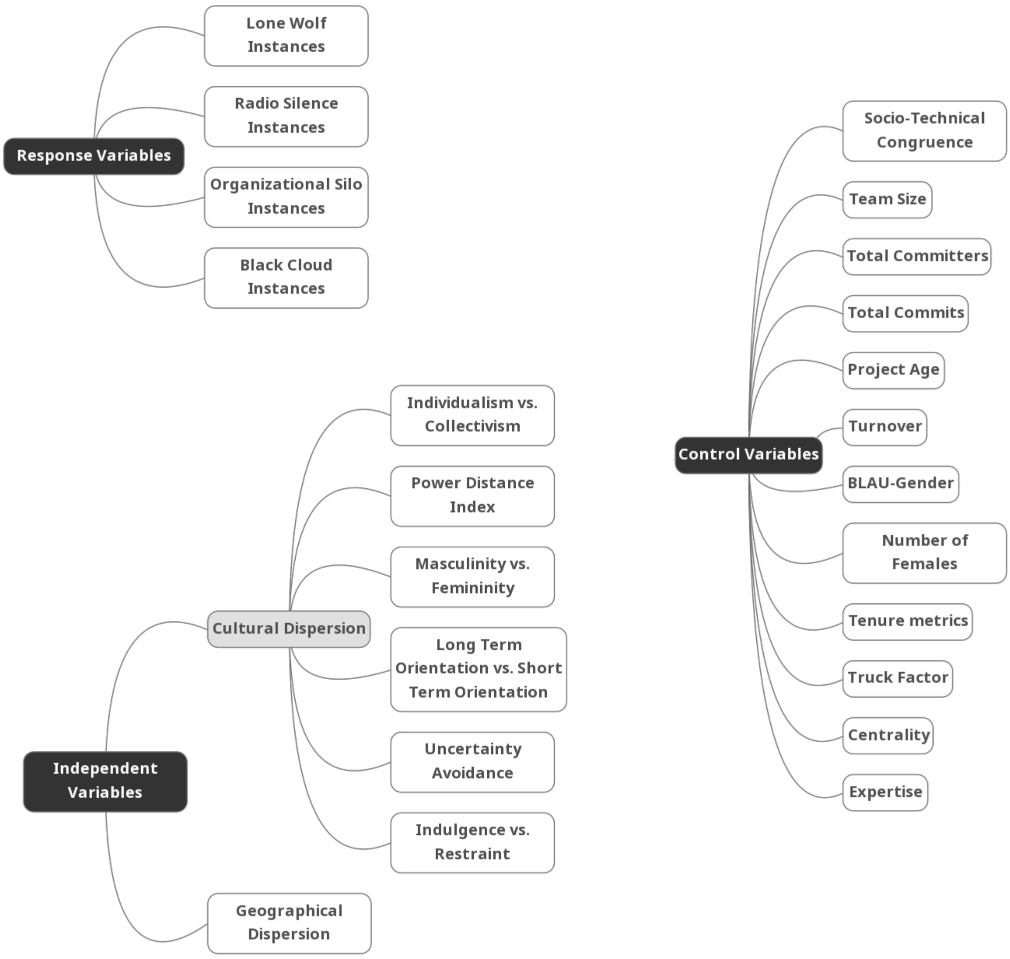

Figure 3.6: Models' variables summary.

computational notebook [109], that is, *Jupyter Notebook*[7] for Python and *R Markdown*[8] for R. The eight obtained models, four from the `statsmodel` Python library and four from the `lmer` R package, showed similar results. For that reason—even though we will provide the scripts for both the technologies—we will show only the models obtained using the R package. Figure 3.6 summarizes the models' components.

[7] `https://jupyter.org/`

[8] `https://rmarkdown.rstudio.com/`

CHAPTER 4

ANALYSIS OF THE RESULTS

This chapter illustrates the achieved results for each research question and the conclusions regarding the main objective.

4.1 $\mathbf{RQ_1}$ - Culture and Geography of GitHub

$\mathbf{RQ_1}$. Which are the cultural and geographical characteristics of open-source communities and developers on GitHub?

As we have said in Section 3.4, the aim of **$\mathbf{RQ_1}$** has been to obtain insight into the cultural and geographical characteristics of open-source communities and their members on GitHub.

Figure 4.1 reports the percentage of developers on GitHub for each country. For legibility reasons: (1) we omitted the percentage inferior or equal to 1%, and (2) we grouped under "Others" label all countries with a percentage inferior or equal to 0.5%. North American countries: the U.S. and Canada count for 43% of developers. The European countries: United Kingdom, Germany, France, Netherlands, Sweden, Spain, Poland, Switzerland, Italy, Ukraine, and Norway, count for 29% of developers. Brazil, Australia, Russia, India, Japan, and China count for 12% of developers. Countries with

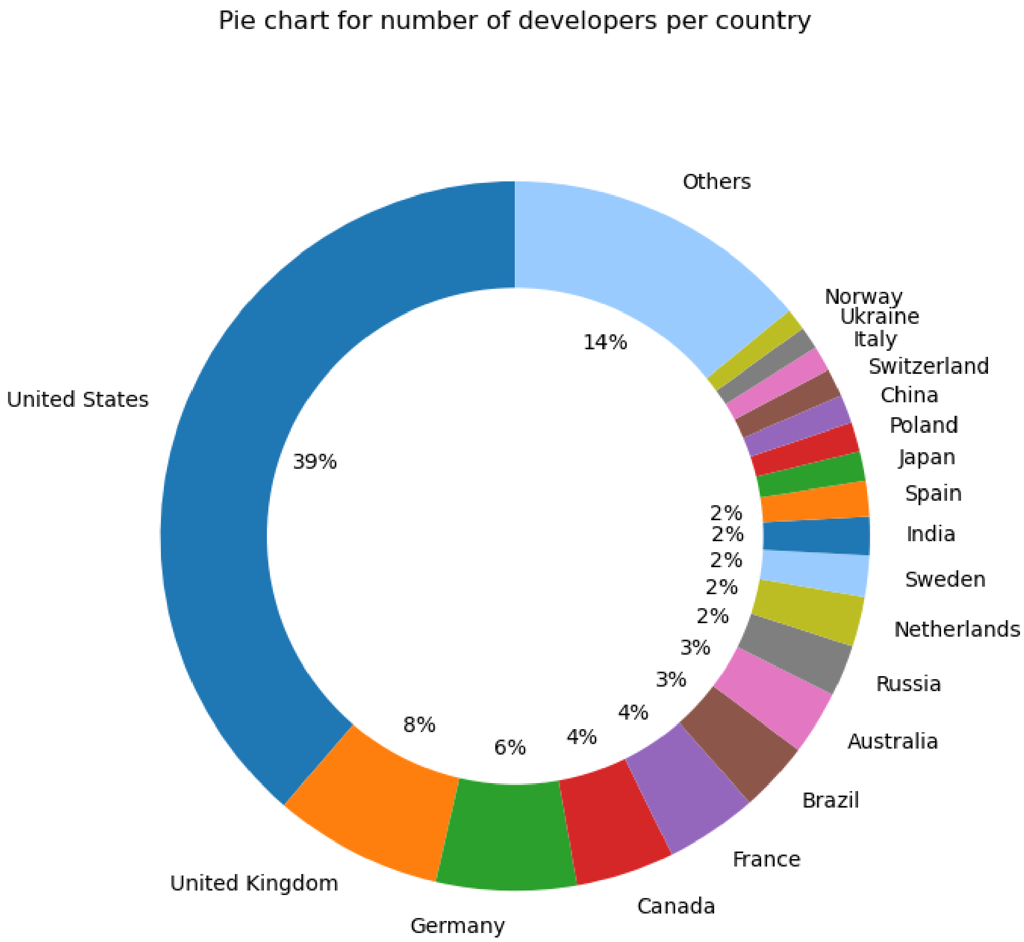

Figure 4.1: Top countries according to percentages of developers.

an individual percentage inferior or equal to 0.5% grouped count for 14% of developers. In Figure 4.2, we report the same information discussed above but using numbers instead of percentages. As shown in the two figures mentioned above, the vast majority of the developers on GitHub come from the United States of America—as confirmed by others studies [110, 111]. This result is significant since this imbalance will influence most of the other results derived from quantitative studies.

As the second part of our study, we have analyzed the cultural characteristics of all the developers in our dataset. To do so, we have used Hofstede's dimensions [12] to characterize cultural behavior. We have widely

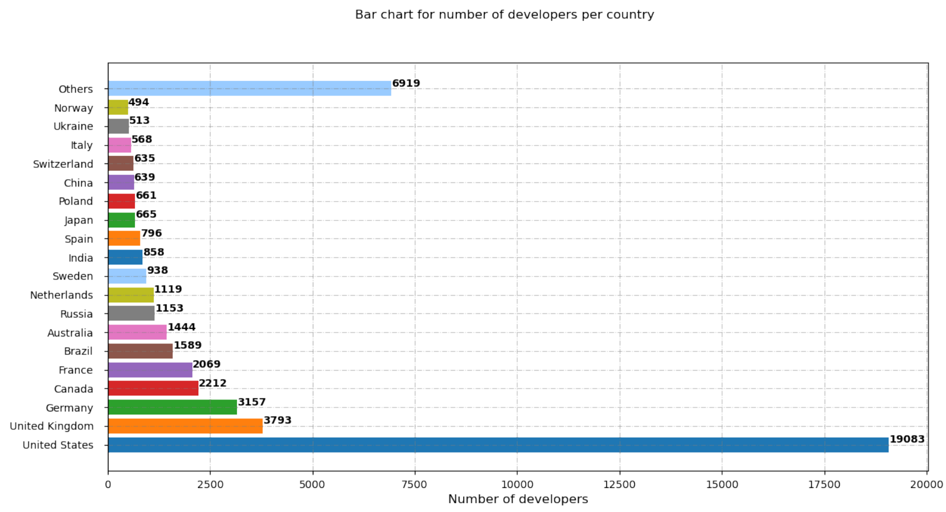

Figure 4.2: Top countries according to number of developers.

discussed Hofstede's 6-D framework in Section 2.2.2; nevertheless, in Table 4.1, we have reported such dimensions with a brief definition to facilitate reading.

Focusing on the cultural analysis of open-source developers, Figures 4.3 reports the developers' values for each Hofstede measure. We choose to use violin plots in order to represent also the dispersion and the differences between the various developers. Causing that Hofstede's dimensions are based on countries, the averages values—the green ones in the figure—for each measure are very close to the United States ones—as anticipated above. An interesting note resides in the fact that the countries with most of the developers (and for that, also the developers) present very similar values for five dimensions, that is, *Power Distance*, *Individualism*, *Masculinity*, *Uncertainty Avoidance*, and *Indulgence*. We can say that because in the violin plots for such dimensions, the bulge of the graphs, representing the frequency of the values, is positioned near the median. The only measure that shows more dispersed values is *Long Term Orientation*. We can conclude that, except for *Long Term Orientation*, open-source developers tend to have similar cultural ideas and attitudes.

Finally, in Table 4.2, we report the standard deviation and mean of all Hofstede's dimensions in the developers' dataset. We have used standard deviation to show how culturally different developers on GitHub

Table 4.1: Summary on the Hofstede's dimensions.

Dimension	Definition
PDI	Power Distance Index expresses the degree to which the less powerful members of a society accept and expect that power is distributed unequally.
IDV	Individualism vs. Collectivism represents the degree to which people in a society are integrated into groups.
MAS	Masculinity vs. Femininity refers to the distribution of roles between men and women.
UAI	Uncertainty Avoidance expresses the degree to which the members of a society feel uncomfortable with uncertainty and ambiguity.
LTO	Long Term Orientation vs. Short Term Orientation measure how much people are oriented toward a long-term outlook in contrast to a more short-term.
IVR	Indulgence vs. Restraint refers to the degree of freedom that societal norms give citizens to fulfill their human desires.

are synthetically—it can assume values from zero to fifty. Specifically, a value close to zero means low dispersion, while close to fifty means high dispersion. For instance, if we have a set of four developers, the first two with a dimension value of zero (the minimum) and the last two with a value of one hundred (the maximum), the standard deviation will be fifty (max dispersion). As is shown in Table 4.2, the statistics for all Hofstede's dimensions are low, with a minimum of 16.064 for *Power Distance Index* and a maximum of 21.523 for *Long Term Orientation vs. Short Term Orientation*. We can consequentially conclude that developers on GitHub are not much culturally dispersed, except for the *Long Term Orientation* dimension.

Table 4.2: Means and Std. Dev. of developers' Hofstede's dimensions.

Statisitc	PDI	IDV	MAS	UAI	LTO	IVR
Mean	45.837	74.735	55.907	55.279	43.860	58.499
Std. Dev.	16.064	20.943	15.874	18.705	21.523	16.450

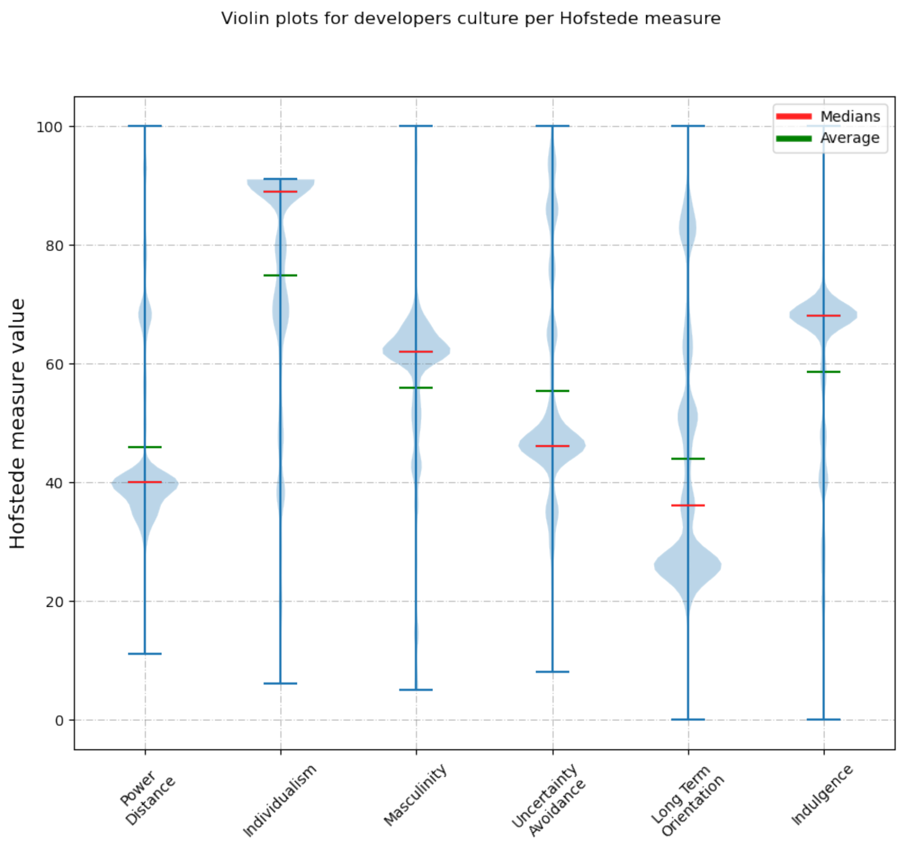

Figure 4.3: Hofstede's dimensions values for developers.

4.2 RQ$_2$ - Cultural Dispersion of GitHub Communities

RQ$_2$. How much are the development communities on GitHub culturally dispersed?

As we have said in Section 3.5, at the communities level, we have studied (1) the cultural average values of development communities and (2) how much the communities are culturally dispersed internally (that is, how members of the same community tend to have different opinion regarding social aspects).

Figure 4.4 reports the average of each community per Hofstede measure. It is possible to note that, in contrast with the developers' cultural representation in Figure 4.3, development communities presents more variety in their average culture value. Also, in that case, the Hofstede dimension with the majority of different values is *Long Term Orientation*. We can also notice

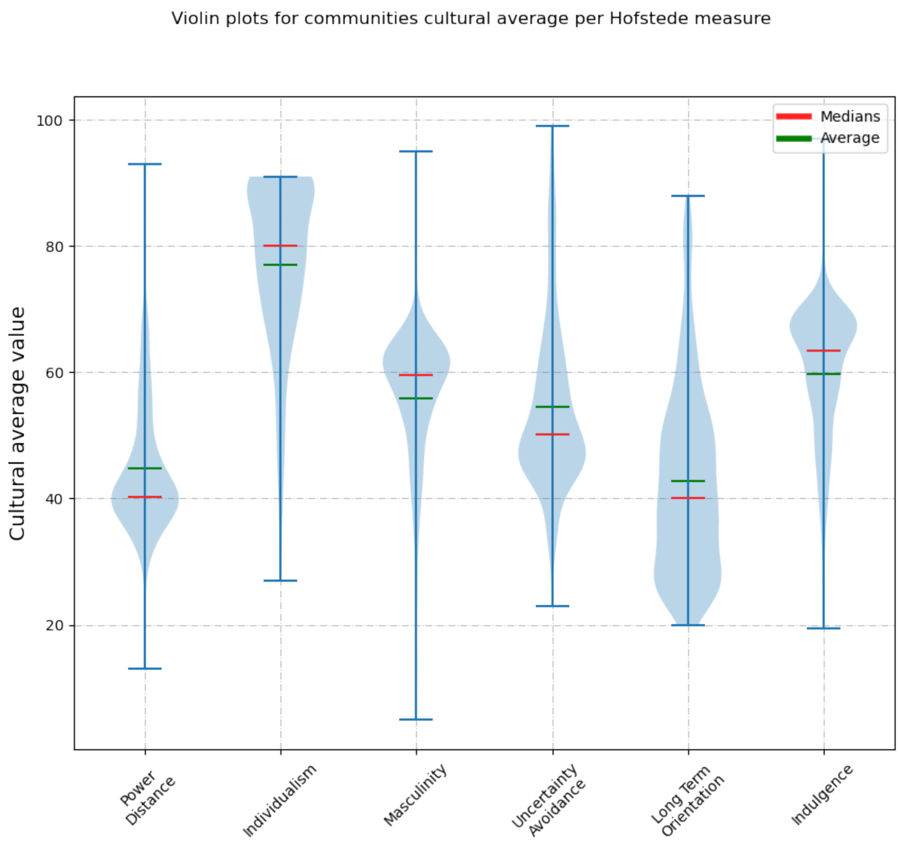

Figure 4.4: Hofstede's dimensions average values for communities.

that the means and medians of each violin plot are near the Hofstede measure values of the United States because of the significant presence of open-source developers coming from such country on GitHub. We can conclude that development communities differ in their members' opinions regarding investing or not for the future and maintaining old traditions.

Figure 4.5 reports the standard deviation of each community per Hofstede dimension. We have used such statistics to operationalize cultural dispersion in development communities. It is possible to note that the majority of the communities present zero (or near zero) cultural dispersion (as also represented by the fact that the means and medians of all violin plots are near zero value). The previously obtained result means that the community members tend to have the same idea on almost all cultural aspects. Also, in that case, the measure that presents more dispersed values is *Long Term Orientation*, meaning that member of the same community tends to have different ideas

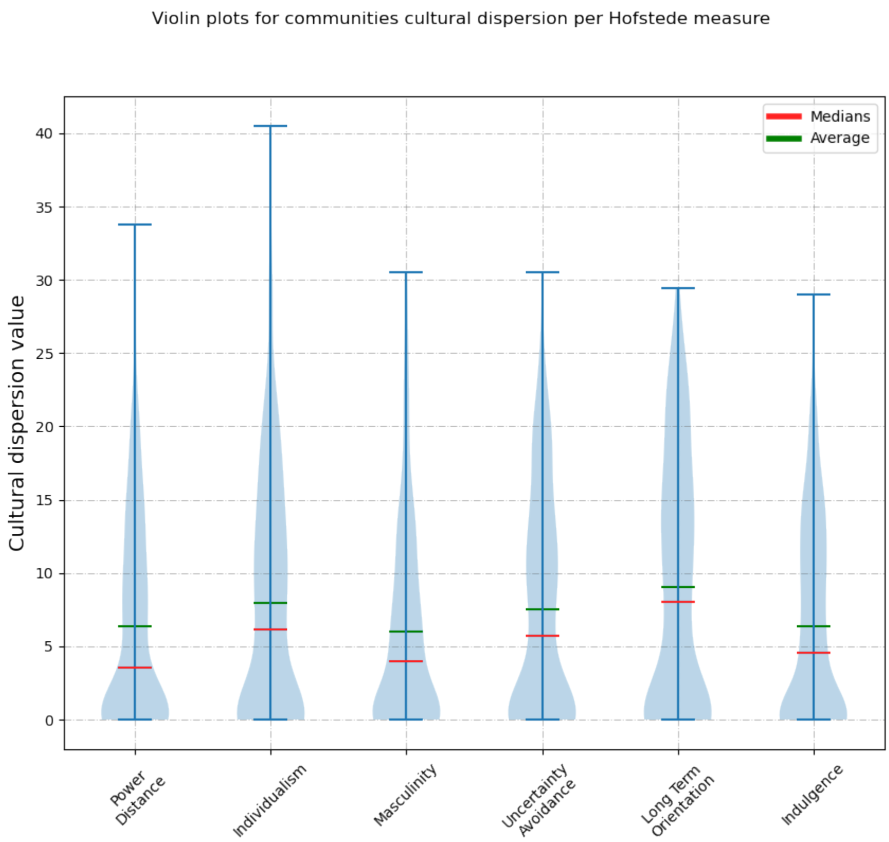

Figure 4.5: Cultural dispersion values for communities.

regarding investing or not for the future and maintaining old traditions.

4.3 RQ$_3$ - Cultural Variation in Time

RQ$_3$. In which way do cultural characteristics of development communities change over time?

As we have said in Section 3.6, with the third research question we aimed to verify if, with the continuation of the development project, the communities tend to attract people from different or the same cultures of the developers already in the team.

Figure 4.6 reports the values of the cultural variation for the development communities selected, per dimension of Hofstede. From the figure, we can notice that the mean and median of the cultural variations of all communities are near zero, which means that development communities tend to do not

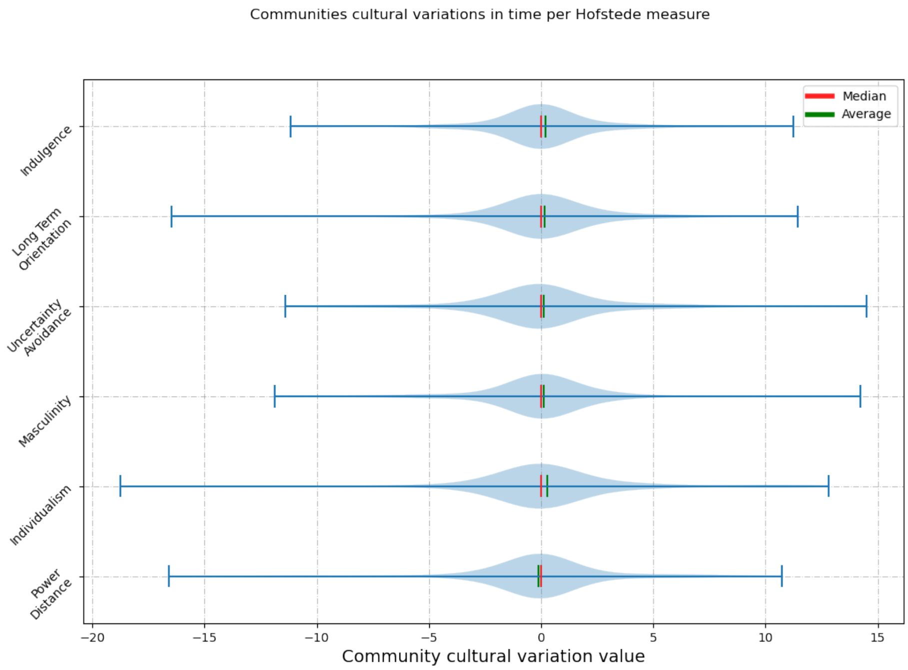

Figure 4.6: Cultural variation for communities.

change with the project's progression. Also, the outliers values are not distant from the zero value and are not in large numbers, and for that, they do not influence the obtained result very much.

Figure 4.7 reports the values of the standard deviations of the communities' cultural variations per dimension of Hofstede. It is a metric able to tell us if the variations of all the communities' instants pairs are similar or not. From the figure, we can notice that the mean and median of the values are near zero (the more distant value is eight), which means that the variations of the instants pairs are similar. In that case we can also see some outliers far away from the median and median. We can conclude that development communities tend to attract people from the same culture or with similar ideas with respect of the already hired members.

Regardless, this research question has to be taken as an explorative work that concerns the cultural variation measure. Therefore, further research is needed in such a direction.

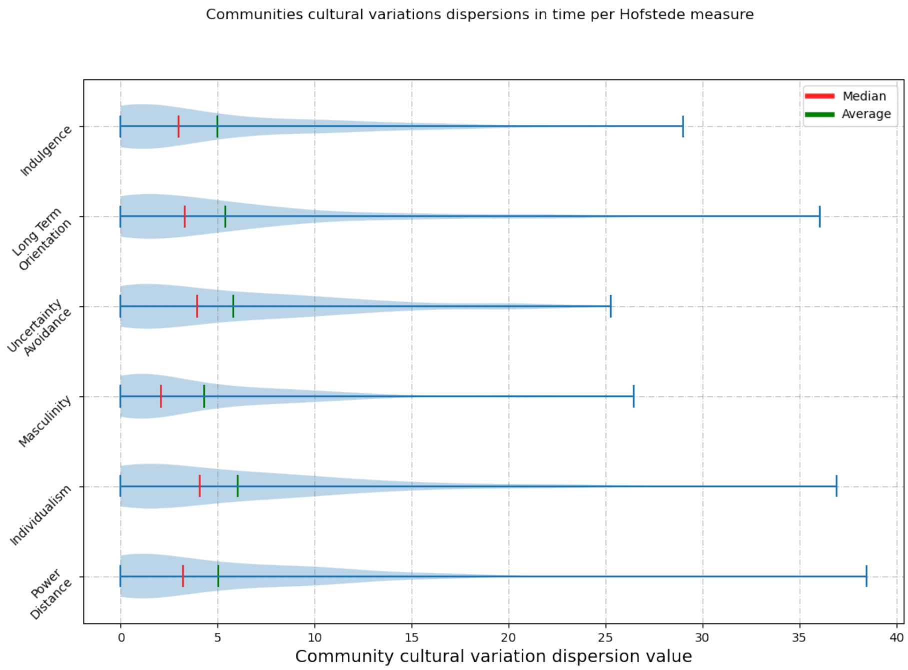

Figure 4.7: Cultural variation dispersion for communities.

RQ_1 - RQ_2 - RQ_3: summary of the results.

Our statistical overview of GitHub developers and their communities reveals that such a platform features a vast majority of people in the United States. In addition to this, development communities tend to attract people from the same cultural group as already recruited members, thus disadvantaging cultural diversity.

4.4 RQ_4 - Focus Group with Open-Source Developers

RQ_4. In which way do cultural and geographical dispersions impact collaboration and communication activities in open-source communities?

As we have said in Section 3.7, we performed a focus group to address our questions qualitatively. We intended to enable a joint discussion on the cultural

Table 4.3: Background of the focus group participants.

ID	Role	Context	Dev. Exp.
P1	Project Manager	Software Development	19
P2	Senior Solution Architect	Software Development	21
P3	Senior Quality Engineer	Research	10
P4	Project Manager	Software Development	15
P5	Software Engineer	Software Development	10
P6	Quality Engineer	Research	10

and geographical differences of the developers inside development communities and their impact on communication and collaboration activities. We invited six developers with experience in distributed teams and asked them questions that have the aim of:

1. discovering if Hofstede's dimensions correctly represent the cultural attitudes in the development communities;
2. exploring the more present cultural differences in development communities and their influence on the emergence of communication and collaboration problems;
3. gathering some insights on possible refactoring and mitigation strategies for such culturally originated problems.

In the following sections, we have reported the main results. To maintain the anonymity of the focus group participants we have referred to them using identifiers shown in Table 4.3.

4.4.1 Hofstede as Cultural Model for Software Engineering

After the introductory part, the first asked question regards the developers' perception about the capacity of Hofstede's dimensions to represent cultural attitudes in development teams correctly. All participants agreed that, based on their experience, all the presented cultural dimensions significantly represent the developers' cultural aspects in software communities. In this respect, P1 commented that *"In my experience, I certainly experienced people that fit each of the six categories of Hofstede's dimensions."*. P2 confirmed this line of thinking and added that *"Most of the conflicts arise because people tend to do things in a certain way, and the mentioned dimensions describe those types of characteristics very well."*. According to these opinions, we can assert that Hofstede's dimensions are capable of representing culture in development teams.

4.4.2 Cultural and Geographical Differences Impact on Communication and Collaboration

In this section, we have summarized the main discussed topics during the core part of the focus group. For the sake of readability and clarity, we have decided to divide the obtained results by the main discussed and emerged topics. Furthermore, we separated each topic into two parts: (1) the core of the discussion, regarding the cultural difference, and (2) the proposed mitigation and refactoring strategies.

Individualism vs. Collectivism

One of the most experienced cultural differences has been *Individualism vs. Collectivism*. More specifically, our participants said that it is prevalent to have both individualistic and collectivistic people in the same team. In that sense, P1 pointed out, *"the most challenging part for me is to face with individuals who prefer to work in isolation on their own things."*. Undoubtedly, a team composed of both individualistic and collectivistic people tends to be characterized by some problems. P1 said that *"There was definitely an issue because individualistic individuals constantly had friction, working with other individuals."*. Always the same participant, talking about a specific event, added that *"I tried to pair them up with teams that tried to come up with projects where collaboration would intentionally be needed in the hopes that they would get more comfortable with it. It worked to a certain degree, but I found that the problem with that is that it dragged up productivity for the other individuals involved because I found that they were overcompensating to make up for the fact that this individual wanted to work in isolation. So yeah, I would say it definitely impacted collaboration communication as a whole, in my attempt to try and fix it for this one individual, I found that it was dragging on performance for the other individuals were being brought into the mix."*. It is clear that not only does such a difference originate communication friction between community members, but that trying to integrate these individualistic individuals into a team leads to the emergence of collaboration problems.

For which concern some mitigation strategies for problems originated by *Individualism vs. Collectivism* differences, it seems that supporting the individualistic nature of such individuals can lead to a practical benefit if combined with a correct division of responsibilities. P1 says that *"In terms of strategies that I ended up using, I then redefined the responsibilities for the individual, where I made it so that a bulk of his work happens in isolation."*. At this assertion, P6, talking about one of his individualistic team members, added that *"he is still working on his own, most of the time, he is actually doing a really good job on his own, and I think it is better not to take him too much into the group stuff, because he has much better results when he is working on his thing, and nobody really bothers him that much."*. In addition to that, another mitigation strategy consists of being present during all the communication phases as a manager to moderate the conversation. In this regard, P1 said that *"In other cases where they need to collaborate, either I will be there as a function of my job as a manager, or that they will be insignificant enough that there should not be that much conflict between the person that they are collaborating with and the approach that they want to take."*.

Summary of the results: Individualism vs. Collectivism. The presence of individualistic people in development teams seems to lead to the emergence of conflict and other problems. The preferred strategy to avoid problems that originated from the discussed cultural difference is to support the developers' individualist nature, leading to the born of a lone wolf, "positive" at least for the sake of the project completion.

Cultural and Geographical Dispersions

Discussing the physical distance between developers, P3 pointed out that *"When we communicate with people from a very distant culture and geographically distant from us, we need to repeat the question five times to actually find what the guy is asking. Similarly, the answer should be repeated five times. Moreover, if they say yes, you need to verify ten times that it means - Yes, I will do it - and not just - Yes, you do it on your own."* To that, P4 added that *"A certain cultural environment indeed poses you in perspective to answer some questions or behave in a certain way, and that is one case where conflict arises because you are used to understanding things in a certain way, behaving a certain way, and then the person you have in front of you starts behaving differently."*. In the end, P1 specifies that *"I find that some cultures tend to be less aggressive than others. If they want to say something negative, they will put it in a better way. In comparison, other cultures tend not to do that right; They will just get directly to the point. However, I find that in the latter case, sometimes if you have someone like that, and he is communicating through email, I think it exacerbates the problem rendering it as more negative than it is really intended to be."*. These observations lead us to the argument that combining cultural differences with distributed work significantly increases the emergence of communication and collaboration problems. Interestingly, most of these problems seem to arise from trivial misunderstandings and lead to communication overhead.

No refactoring and mitigation strategy was proposed by our participants other than being repetitive and making sure over and over again that the discussed topic is understood.

> ***Summary of the results: Cultural and Geographical Dispersions.*** Communication and collaboration from people from different cultures can undoubtedly be more difficult if performed using remote devices. It seems that, in such a context, some cultural behaviors could be easily misunderstood and mistaken for offensive attitudes. Furthermore, the physical distance between community members seems to discourage clarifying approaches to avoid the aforementioned problems.

Uncertainty Hiding

Another interesting discussion arises around the conflict between western and eastern cultures. P2—which has few years of experience in a distributed team with people from both cultures—said that *"I worked in software development teams with a mixed culture from western and eastern cultures and in the last one people do not like risky situations, they do not like potential conflicts to the point where they try to sweep things under the rug or to do anything possible to avoid that the conflict from arising, sometimes at the peril of the whole situation."*. On the consequences of such an attitude, always P2 continue to say, *"So basically, if there is a deadline that is not possible to be met at all, they might wait until the last moment, until the last day, for communicating the fact, because they tend to try to think that - it is okay, it will somehow rule itself out - and that is, that occurred in my practice more than once."*. Reasoning around these statements, it seems that how both cultures treat risk can lead to the emergence of communication and collaboration problems. Talking about the risk, we thought that such difference could be linked to the Hofstede's dimension known as *Uncertainty Avoidance*, namely, "the extent to which the members of a culture feel threatened by ambiguous or unknown situations" [12]. To confirm our thinking, we searched acceptance by P2. Interestingly, P2 precise that *"Well, it is a bit different. It was that people on the western side were fine in finding out basically mapping the risk map, making the risk map ... tackling risks in a proactive manner, rather than the reactive manner ... what we have seen on the eastern side is that there was also a bit of productivity in taking risks, but it was much lower, and mostly it was rather about avoiding and consuming risks on their own. So if there is something going wrong or about to go wrong, it is that person's problem, that group's problem. So they do not expose it internally and ask for help."*. From this statement, we understood that the difference is not in the way people approach risk—that is, *Uncertainty Avoidance*—but in the way people expose it. For that reason, we decided to name this cultural attitude ***Uncertainty Hiding***, namely, the extent to which the members of a culture feel comfortable exposing and revealing risks to ask for help instead of managing them by themselves. Of course, ulterior research is needed to explore such an attitude better and validate it as effectively separate from *Uncertainty Avoidance*, to which is certainly related.

For which concern some mitigation and refactoring strategies for problems originated by differences in the way people from different cultures make the risk clear, P2 pointed out that *"Basically, there was a strike force to find out aspects where the cultures are different, and figure out the solutions for them and turn this into the kind of a handbook, or book of best practices."*. This statement reveals that an exploration-plus-mentoring strategy could be effective in the aim of refactoring such problems. Furthermore, P2 added that *"The second best thing that we did was establish a consistent, continuous training program for everyone arriving on-site or who was starting to interact with the partners and with customers in the other country ... and that really helped in mitigating many problems in our collaboration."*. This statement strengthens the use of mentoring and training strategies as a mitigation approach to such problems.

Summary of the results: Uncertainty Hiding. Some cultures, particularly the eastern ones, seem to expose risks and risky situations differently if confronted with the others. This discovery has led us to define a new cultural dimension, namely, *Uncertainty Hiding.* When conflicts arise in such a cultural attitude, an exploring and mentoring strategy is needed to avoid the emergence of catastrophic problems.

Uncertainty Avoidance: Risk Tolerance

As we have mentioned previously in this section, the Hofstede's dimension known as *Uncertainty Avoidance* indicates the extent to which the members of a culture feel threatened by ambiguous or unknown situations. This dimension is highly related to the risk tolerance of people: a high level—near to one hundred—indicates that people are less risk-tolerant; on the other side, a low level—near to zero—indicates that people like risk and risky situations. On that line, P5 said that *"In my experience, what I noticed was that there is synergy. So it is a pretty big advantage by having people with the high-risk tolerance and the low-risk tolerance on the same machine or the same team."*. From the previous assertion, we can assume that having both the typologies of people (high-risk tolerance and low-risk tolerance) could lead to benefits in the workplace. On such benefits, P6 added that *"I happen to be generally a very high-risk-tolerant person, like my immediate manager. In my experience, this does have the advantage of helping us sort of forge forward and work on larger projects without getting too bogged down. But it certainly helps to have more lower-risk people around, acting to sort of rein in our own natural tendencies and sort of to see things that, by temperament, we might tend to overlook."*. Of course, the fact that a balance in risk tolerance could lead to benefits is well known in modern state of the art [112]. However, in this specific case, the exciting thing is that such a risk tolerance attitude is related to cultural behavior, so at education at a young age, and this means that, in some way, it could be conveyed.

In the specific case of *Uncertainty Avoidance*, specifically in the risk tolerance attitude, it seems that communication and collaboration problems do not emerge. On the contrary, it is clear that having such a typology of dispersion could prevent the emergence of problems, also not connected to communication and collaboration.

Summary of the results: Risk Tolerance. The fact that having a good balance between people with a high-risk tolerance and a low-risk tolerance could improve the project development cycle quality is well known in the project management field modern state of the art. To this knowledge, we can add that such an attitude of tolerance to risk changes from culture to culture.

Uncertainty Avoidance: Gender Disparity

In the end, our conversation moves on the topic of gender disparity due to cultural influences. We categorize gender disparity under *Uncertainty Avoidance* because this dimension measure also the degree to which a culture tends to be more flexible, open to new ideas, and less traditional. One might think that the correct dimension for this attitude is *Masculinity vs. Femininity.* However, the latter refers more to community-preferred behaviors rather than the tendency to treat males and females differently. About this topic, P2—talking about an experience with two sub-communities geographically distributed in different and distant places—said that *"We had the developers on one side and the testers on the other, and the head of testing, the head of QA, was a woman, and the other culture is not very accepting of women in the workplace. So, obviously, that has been a huge problem for us because we have had quite a few females now our team exceptionally talented, and the other culture has been completely unreceptive, they were completely not understanding how those people could be in the same room with them."*. P2 also specified that this was very specific for his experience, so ulterior research is needed. Despite this, we can hypothesize that the simple presence of people from cultures that treat gender roles differently could lead to communication and collaboration problems. Undoubtedly, the presence in such teams of people of different gender could exacerbate the problems and lead to conflicts.

For which concern gender disparity, as a refactoring strategy P2 pointed out that *"The only way we managed to tackle that was to literally move around a few people on both sides, mostly on the other side, so that we manage it to find the right interaction. So basically, we found more flexible and less rigid people who managed to work with our female workforce."*. From the previous assertion, we can say that a possible strategy is to explore the "rigid" community to find less rigid people willing to collaborate with the other gender. The interesting—and, of course, alarming—thing that emerges from what P2 said is that this has been the only way to address such a problem.

Summary of the results: Gender Disparity. The fact that gender disparity is a serious problem, not only in the IT field, is well known in most of the world. Our qualitative study confirms such knowledge but contextualizes it as a cultural problem. Moreover, having people in a team that encourage gender disparity leads to collaboration problems, for example, morale and productivity reduction. An explorative approach, to finding less rigid people willing to collaborate with the other gender, could be effective in refactoring such problems.

RQ_4: summary of the results.

The focus group confirmed cultural and geographical differences as key causes of collaboration and communication problems that threaten the successful completion of software projects. Participants assessed Hofstede's dimensions as a good representation of the cultural differences in software teams. Perhaps more importantly, our results highlighted three key challenges that managers have to deal with: (1) dealing with both individualistic and collectivistic people, (2) correctly balancing risk tolerance, and (3) avoiding gender disparity.

Table 4.4: Overview on community smells used in our study.

Community Smell	Definition
Organizational Silo	Siloed areas of the development community that do not communicate, except through one or two of their respective members.
Black Cloud	Excessive information overload due to a lack of structured communication or cooperation governance.
Lone Wolf	Defiant contributor who apply changes in the source code without considering the considering the opinions of her peers.
Radio Silence	One member interposes herself into every interaction across sub-communities.

4.5 RQ$_5$ - Building a Statistical Model

> **RQ$_5$**. To what extent do cultural and geographical dispersions within teams influence the number of community smells?

In this section, we report the results that address the **RQ$_5$**. We intended to build four mixed linear models, one for community smells types, to study the correlation between cultural metrics—plus others control factors—and the number of smells in a development community. In Table 4.4,we report a summary of the four smells involved in this study. We use both Python and R libraries to build the models mentioned above. Considering that both the technologies created very similar models, we report only the results obtained using the R package `lmer` in this section.

To better analyze and interpret our findings, we build other two baseline statistical models: the first one containing all the control factors and the random effect, but not the independent factors; the second one containing only the random factor. After that, we compared the models with the corresponding baselines using the *AIC* (*Akaike information criterion*) and *BIC* (*Bayesian information criterion*) [113, 114]. *AIC* and *BIC* are estimators of prediction error and quality of statistical models for a given data set. These metrics estimate the quality of each model in order to provide a means for model selection [114]. The general rule is that the model with the lower AIC and BIC is the one that better characterizes the sample. The comparison with the first

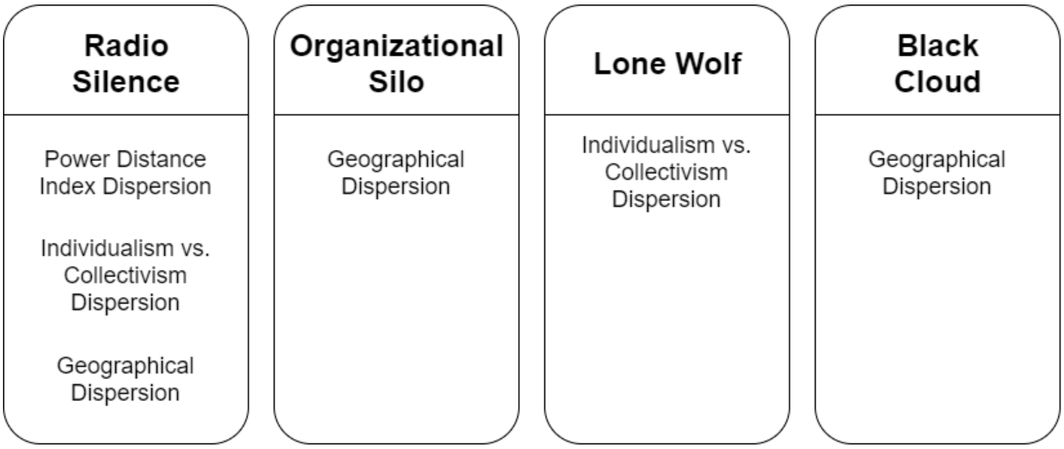

Figure 4.8: Community smells with the correlated independent variables.

model allowed us to study how the control variables, without the independent factors, influence the number of community smells. The comparison with the second model, the one with only the random factor, allowed us to understand whether the obtained results reflected the random effect.

Figure 4.8 summarizes the results, showing for each smell the correlated independent variables according to the model. In the following sections, we discuss the obtained results separately for each type of smells.

4.5.1 Radio Silence Model

Table 4.5 reports the details regarding the statistical model built for the *Radio Silence* effect. As we can see, software community members' *turnover* is a potent mediator over the occurrence of the target community smell, as implicitly revealed by Homscheid et al. [115]. The second most important factor is the *Blau Diversity Index*, indicating that the presence of women in development teams can influence the number of community smells, as evidenced by Catolino et al. [69].

As regards the cultural aspects:

- *Power Distance Index Dispersion* impacts the response variable with an estimate of 0.07 and a standard error of 0.031 with significance below 0.052;

Table 4.5: Results achieved by the three models built for *Radio Silence*.

Factor	All Variables			Conf. Variables			Random Variable		
	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.
(Intercept)	-4.042	1.025		-1.926	0.645		2.261	0.143	
log(totalCommitters)	-0.091	0.087		-0.093	0.086				
log(totalCommits)	0.101	0.066		0.024	0.061				
projectAge	-0.015	0.014		-0.027	0.014	.			
turnover	9.857	0.471	***	9.982	0.471	***			
blauGender	4.011	1.528	**	0.492	0.938				
tenureMedian	0.067	0.034	*	0.049	0.035				
tenureDiversity	0.008	0.024		0.011	0.025				
log(teamSize)	0.304	0.156	.						
stCongruence	-0.336	0.178	.	-0.391	0.185	*			
truckFactor	-0.018	0.044		-0.012	0.046				
numberOfFemale	0.001	0.022		0.042	0.012	***			
expertise	0.066	0.195		0.054	0.203				
centrality	-0.118	0.142		-0.074	0.147				
PDID	0.071	0.028	*						
IDVD	-0.055	0.023	*						
MASD	-0.001	0.023							
UAID	-0.007	0.029							
LTOD	0.036	0.024							
IVRD	-0.013	0.037							
GeoD	-0.001	0.001	*						
Legend for Sig.	$*** : p < 0.001$; $** : p < 0.01$; $* : p < 0.05$; $. : p < 0.1$								

- *Individualism vs Collectivism Dispersion* impacts the response variable with an estimate of -0.052 and a standard error of 0.021 with significance below 0.051;

- *Geographical Dispersion* impacts the response variable with an estimate of 0.012 and a standard error of 0.001 with significance below 0.047.

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

4.5.2 Organizational Silo Model

Table 4.6 reports the details regarding the statistical model built for the *Organizational Silo* effect. As we have also seen in the previous section,

Table 4.6: Results achieved by the three models built for *Organizational Silo*.

Factor	All Variables			Conf. Variables			Random Variable		
	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.
(Intercept)	4.997	1.497		2.419	1.216		2.286	0.123	
log(totalCommitters)	0.402	0.153	**	0.423	0.159	**			
log(totalCommits)	-0.218	0.122	.	-0.215	0.118	.			
projectAge	-0.024	0.026		-0.033	0.027				
turnover	-1.601	0.561	**	-1.738	0.571	**			
blauGender	0.414	2.066		2.514	1.801				
tenureMedian	-0.066	0.064		-0.057	0.067				
tenureDiversity	0.039	0.044		0.028	0.047				
log(teamSize)	0.133	0.157							
stCongruence	0.691	0.332	*	0.531	0.351				
truckFactor	0.025	0.081		0.046	0.087				
numberOfFemale	0.017	0.021		0.014	0.022				
expertise	-0.469	0.367		-0.462	0.391				
centrality	0.086	0.263		-0.084	0.274				
PDID	0.001	0.052							
IDVD	0.008	0.043							
MASD	-0.033	0.041							
UAID	0.013	0.057							
LTOD	-0.023	0.044							
IVRD	-0.023	0.071							
GeoD	-0.001	0.001	***						
Legend for Sig.	$***: p < 0.001$; $**: p < 0.01$; $*: p < 0.05$; $.: p < 0.1$								

Turnover is an influential factor in the emergence of the smell. Also, the total *number of committers* influence the number of smells significantly. We can explain the two results mentioned above because a high number of committers combined with an exceedingly static community (that is, that does not change very much) can lead to a division in consolidated sub-communities.

Regarding the cultural aspects, *Geographical Dispersion* seems to impact (negatively) the number of *Organizational Silo*, with an estimate of -0.001 and a standard error below 0.001 with significance below 0.001. This result indicates that the physical nearness of the developers can lead to the emergence of sub-communities.

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

Table 4.7: Results achieved by the three models built for *Lone Wolf*.

Factor	All Variables			Conf. Variables			Random Variable		
	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.
(Intercept)	-9.123	4.382		8.756	2.292		8.188	0.346	
log(totalCommitters)	0.361	0.297		0.226	0.311				
log(totalCommits)	0.216	0.232		0.118	0.231				
projectAge	0.093	0.051	.	0.123	0.053	*			
turnover	0.724	1.093		1.285	1.124				
blauGender	-4.044	3.758		-0.933	3.543				
tenureMedian	-0.052	0.121		-0.008	0.131				
tenureDiversity	0.045	0.085		0.065	0.093				
log(teamSize)	4.777	0.877	***						
stCongruence	-7.654	1.503	***	-14.092	0.943	***			
truckFactor	-0.221	0.156		-0.201	0.171				
numberOfFemale	0.004	0.037		0.016	0.041				
expertise	0.784	0.687		0.661	0.749				
centrality	0.078	0.511		0.958	0.535	.			
PDID	-0.162	0.101							
IDVD	-0.194	0.083	*						
MASD	0.001	0.081							
UAID	0.158	0.111							
LTOD	0.047	0.086							
IVRD	0.094	0.135							
GeoD	-0.001	0.001							
Legend for Sig.	$***: p < 0.001$; $**: p < 0.01$; $*: p < 0.05$; $.: p < 0.1$								

4.5.3 Lone Wolf Model

Table 4.7 reports the details regarding the statistical model built for the *Lone Wolf* effect. For this type of smell, *team size* and *socio-technical congruence* seem to be the most relevant factors. Regarding the cultural aspects, *Individualism vs Collectivism Dispersion* impacts the number of smells, with an estimate of -0.194 and a standard error of 0.083 with significance below 0.05. This result can mean that poorly coordinated teams tend to exhibit more *Lone Wolves* instances, but that team members with different ideas about the concepts of individualism and collectivism can mitigate their emergence.

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

Table 4.8: Results achieved by the three models built for *Black Cloud*.

Factor	All Variables			Conf. Variables			Random Variable		
	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.	Estimate	S.E.	Sig.
(Intercept)	7.266	1.899		5.923	1.575		4.466	0.177	
log(totalCommitters)	-0.011	0.195		-0.046	0.211				
log(totalCommits)	-0.018	0.156		0.017	0.156				
projectAge	-0.005	0.032		-0.017	0.035				
turnover	0.785	0.730		1.127	0.762				
blauGender	-7.668	2.708	**	-10.906	2.412	***			
tenureMedian	0.042	0.081		0.025	0.088				
tenureDiversity	-0.001	0.057		-0.005	0.063				
log(teamSize)	0.171	0.198							
stCongruence	0.249	0.418		0.003	0.464				
truckFactor	0.028	0.104		-0.041	0.115				
numberOfFemale	0.017	0.025		0.035	0.026				
expertise	-0.249	0.464		-0.331	0.514				
centrality	-0.276	0.333		-0.482	0.362				
PDID	-0.004	0.066							
IDVD	-0.021	0.055							
MASD	0.071	0.053							
UAID	0.035	0.073							
LTOD	-0.016	0.057							
IVRD	0.009	0.091							
GeoD	-0.001	0.001	***						
Legend for Sig.	$*** : p < 0.001$; $** : p < 0.01$; $* : p < 0.05$; $. : p < 0.1$								

4.5.4 Black Cloud Model

Table 4.8 reports the critical pieces of information regarding the statistical model built for the *Black Cloud* effect. As we can see, *Blau-index*, which denotes diversity, has a firm estimate and significance.

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

Regarding the comparison with the baseline, our model achieved: (1) an AIC value lower than the one with the confounding variables and the one with only the random factor and (2) a BIC value equal to the one with the confounding variables and greater than the one with only the random factor.

Then, we can assert that the addition of the independent factors could help explain the response variable better, but that deeper analysis is needed.

RQ_5: Summary of the results.

Cultural Dispersion is a relevant factor in the emergence of *Radio Silence* and *Lone Wolf*, while *Geographical Dispersion* is relevant for all the smells except *Lone Wolf*. In this perspective, it seems that cultural dispersion mainly impacts communication-based smells, compared to structure-based ones. Despite this, ulterior research on the correlation with *Radio Silence* is needed considering the quality metrics compared to the baseline with only the confounding factors.

CHAPTER 5

OBSERVATIONS AND LESSONS LEARNED

This chapter illustrates observations on the research and the lessons learned for the research community.

5.1 Theoretical Convergence

Recall that the aim of our study, specifically in research questions four and five, was understanding the role of cultural and geographical dispersions as factors contributing to the rise of collaboration and communication problems, representable using community smells. To better explore the context of the study and strengthen the obtained results, we adopted a mixed-method-research approach, namely, the combination of qualitative and quantitative techniques finalized to the same objective. The main objective of such a method is to converge to the same theory using different approaches. In order to describe the convergence in question, we used the theoretical lemmas that our research identified, followed by the resulting *Theoretical Convergence*.

From a qualitative perspective, the data obtained during the focus group shows that developers and managers experience cultural dispersion problems, specifically in the opposition between individualistic and collectivistic people. Furthermore, those contrasts lead to the emergence of collaboration and communication problems, especially for the management of team-working and comprehension during informal communication.

> ***Lemma 1.*** Cultural and geographical differences influence communication and team-working.

From a quantitative perspective, statistical modeling indicates that cultural dispersion can influence the number of community smells in open-source communities. Notably, the physical distance between developers and different ideas on individualism and collectivism approaches, as experienced in the qualitative analysis, seem to be the most crucial factors.

> ***Lemma 2.*** Cultural and geographical differences influence the number of community smells in open-source development communities.

Both the strategies lead to very similar results that converge towards the following theory:

> ***Theoretical Convergence.*** Cultural and geographical dispersions influence communication and collaboration problems in software communities, bringing to the emergence of community smells.

5.2 Discussions

The modern state of the art has not yet thoroughly investigated the role of people culture as a factor for software team building and governance. In this thesis, we aimed to add some clarity through both quantitative and qualitative empirical research.

5.2.1 Statistical Overview

A first key discussion point reflects our finding that the communities on GitHub tend to have developers with a similar cultural background and do not change in time. This result is, of course, influenced by the fact that most developers come from the United States. However, it is interesting because GitHub is based on making possible the collaboration of people from different countries and ethnicities, and our results contradict this idea.

5.2.2 Focus Group

Secondly, focusing on the focus group performed to study the impact of cultural and geographical dispersions on communication and collaboration, we can divide the discussion into the four emerged points.

Hofstede in Software Development. First, it is clear that the participants consider Hofstede's dimensions as an excellent way to represent cultural attitudes in development communities. Undoubtedly, this is an ulterior confirmation of the validity of the Hofstede framework adoption in such a context. Said this, Hofstede's dimensions are based on people and their origin countries, so it is a way to represent people's culture. An interesting discussion point for future work could concern ways to define and represent organizational culture for development communities—so the focus shifts from people to the specific company—and how such a culture is determined and influences community members' decisions. It is possible that, in that specific case, the Hofstede framework is not adequate.

Individualism vs. Collectivism. Regarding the cultural attitudes more perceived in the development communities, *Individualism vs. Collectivism* is unquestionably on the top. Having people with different ideas on such an attitude leads to communication and collaboration problems, even though individualistic people are in the minority concerning the counterparty in most cases. The most interesting fact that emerges from the focus group is that trying to integrate such individuals into groups worsens the situation. Thus, the preferred strategy is to support the individualistic behavior and lead to the emergence of "positive" *Lone Wolfs*—wanting to use the same name associated with the corresponding community smells. Undoubtedly, such a decision improves the community members' productivity but could also lead to the emergence of isolation and other correlated problems. However, on the other hand, it can also be possible that the social problems that could emerge from such a choice are less dangerous than the ones from a bankruptcy integration strategy.

Uncertainty Hiding. The second most discussed attitude concern the way people interact with risk and risky situation. More specifically, the discussion regarding how and to which degree people tend to expose risk to the other community members. We have decided to name such an attitude *Uncertainty Hiding*, namely, the extent to which the members of a culture feel comfortable exposing and revealing risks to ask help instead of managing them by themselves. The presence of people with different behavior in *Uncertainty Hiding* can leads to the emergence of communication and collaboration problems. A possible cause could be that people who are used to making problems explicit, when working with others who do not, tend to: first, assume that there are not problems if their colleagues do not speak; second, annoy colleagues who do not understand such an open attitude. Undoubtedly, both situations are dangerous and lead to conflicts. When such conflicts arise, it seems that an exploring and mentoring strategy can avoid catastrophic consequences. However, there is not much to say about that because such *exploration-plus-mentoring* strategies are a well-known standard to resolve comprehension problems, despite the high effort required. Undoubtedly, more research is needed to confirm such a cultural dimension as concretely different from the one known as *Uncertainty Avoidance.*

Uncertainty Avoidance. *Uncertainty Avoidance* monopolized the third part of the focus group largely. In this part, the discussion focused on two main aspects: risk tolerance and gender disparity. For which concern risk tolerance, the focus group confirm that a good balance between people with a high-risk tolerance and a low-risk tolerance could improve the development lifecycle in various ways-as well known in the modern state-of-the-art. Regarding gender disparity, our study confirms such a problem, specifically as a situation in which men are more advantaged than women. Undoubtedly, this is a massive problem because it could be one of the main reasons female workers are less than men. In fact, such a cultural behavior could discourage females, already from a young age, from entering the computer science world, preventing software communities from reaping the benefits of their presence [69].

5.2.3 Statistical Model

Thirdly, focusing on the cultural and geographical dispersions as mediators for community smells, we reported diverse results for each smell.

Black Cloud and Organizational Silo. For example, in the context of *Black Cloud* and *Organizational Silo*, geographically dispersed team members can influence the presence of the smells. For the first smell, a possible motivation is that, to manage people physically located in different parts of the world, it is necessary to use management tools for communication and collaboration, e.g., *Trello* and *Jira*. Of course, we can consider this fact a critical problem for the management world because it is like a way to indirectly oblige tools that should be used as a norm during development. For *Organizational Silo*, probably have team members located nearby can lead to the formation of sub-communities. Like *Black Cloud*, we can consider such a strategy as an indirect and forced way to reinforce standard good practices, such as avoiding excessive sub-communities proliferation.

Lone Wolf. More specifically, for cultural dispersion, we have seen that *Individualism vs. Collectivism Dispersion* is an essential factor for the emergence of *Lone Wolf* effect instances. To better discuss this result is important to remember that most of the developers come from the United States, which has an *Individualism vs. Collectivism* value (for the Hofstede measure) of 91—that is one of the highest values between all the countries. Consequently, people from the U.S. tend to prefer individualistic actions and attitudes. Therefore, we want to say that, in most development communities, a low level of *Individualism vs. Collectivism Dispersion* indicates that team members tend to accept an individualistic attitude, leading to the emergence of *Lone Wolf* instances. On the contrary, a more divergent community tends to enhance integration and mitigate the smell's presence.

Radio Silence. Regarding *Radio Silence*, our model reveals that, as for *Lone Wolf*, *Individualism vs. Collectivism Dispersion* could be a crucial factor. An interesting result is that *Power Distance Index Dispersion*—that refers to the degree to which the less powerful members of a society accept and expect that power is distributed unequally—influence positively the emergence of such a community smell. We can explain this because communities that agree on the assignment of power to individual members are led to leave the communication tasks between the various subcommunities to these individuals.

Control variables. Last, our study confirms the role of established socio-technical quality metrics [99, 78] (e.g., *socio-technical congruence* and *truck factor*) and gender diversity [69] (in the form of *Blau-index*) as mediators for community smells.

5.2.4 Qualitative and Quantitative: Comparison

As said in Section 3.2, for this study, specifically in research questions four and five, to better explore the context and strengthen the obtained results, we adopted a mixed-method-research approach. The focus group acts as the qualitative part of the study, while the statistical model construction acts as the quantitative part. We have already said in Section 5.1 that both the strategies lead to similar results: cultural and geographical dispersions influence communication and collaboration problems, bringing to the emergence of community smells. In this part of the thesis, we want to discuss more deeply the meeting points.

The principal meeting point is the impact of *Individualism vs. Collectivism Dispersion* on the emergence of problems. Specifically, the focus group reveals that such cultural conflict leads to the isolation of individualistic people—also seen as a mitigation strategy to avoid worse problems. The statistical model confirms such a result, showing that the metric under analysis impacts the number of instances of the *Lone Wolf* effect.

The statistical model also confirms the results concerning the *Geographical Dispersion*, showing that such a dispersion could lead to the "proliferation" of *Organizational Silo* effect, *Black Cloud* effect, and—in minor part—of *Radio Silence* effect. On the same line, the focus group results reveal a connection between the *Geographical Dispersion* and the emergence of problems, adding that developers do not have effective mitigation or refactoring strategies. Summarizing, both strategies reveal that physical distance between community members, combined with high cultural differences, could lead to the emergence of problems.

For which concern the impact of *Uncertainty Avoidance*—and of the way people react to risk—, only the focus group reveal a connection with the emergence of communication and collaboration problems. Despite this, it is important to note that such results referred principally to the collaboration of people from western and eastern cultures. Thus, the fact that in the datasets used for the model construction [89, 69], only near the 2% of developers comes from eastern cultures could be the reason for this mismatching. Ulterior research is needed to understand such a correlation better.

5.3 Lessons Learned

In this section, we report some possible lessons learned that could be useful to the IT workers. Specifically, we observed that:

1. **Managers should pay attention to the interactions between individualistic and collectivistic people.** This result is particularly true when the first ones are in very few numbers concerning the second ones. In most cases, encouraging these individuals' individualistic behavior could enhance the productivity of all the team.

2. **People's attitude towards risk is a crucial aspect.** It could be possible that a part of a community disdain risk and even does not want to expose it or talk about it. In such a situation, unexpected and catastrophic events could appear, and the risk management phases acquire even more importance. Moreover, managers should foresee mentoring sessions about the community culture about risk assessment to align all members.

3. **Make a handbook with cultural lessons could be a good idea to mitigate problems related to cultural behavior.** With a view to continuous improvement of the process, preserving what has been learned about the cultures that are part of the community can undoubtedly help to mitigate the most trivial problems. Moreover, training lessons with people that have experienced working with sub-communities culturally different should be implemented.

CHAPTER 6

THREATS TO VALIDITY

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

6.1 Threats to Conclusion Validity

Threats to *conclusion* validity concern the relation between treatment and outcome and are related to issues that affect the ability to draw the correct conclusions at the end of the work.

For the first three research questions, we use descriptive statistics and data visualization techniques to study the culture and geography of GitHub developers and communities. In that case, we use multiple strategies and charts to assure the truthfulness of our results.

The primary threat is related to the statistical method used for the last research question. Firstly, we used two different technologies to conduct our study, i.e., R and Python. To avoid an influence coming from the composition of the dataset, which was composed of multiple time windows for each development project, we chose to use a mixed-effects model [103, 106]. This choice allows us to capture measurements from within the same group, as recommended by the literature. Afterward, we chose to adopt some mitigators for the well-known problem of multicollinearity [107]. Thus, we used the *variance inflation factors* [107] function in the R models, while we used the `statsmodel` library features [104] in the Python models. In addition, we discarded outliers [107] and used only data for which the information on the origin country was available for most developers. Last, to statistically verify the significance of the independent factors' influence on the response variable, we operated ANOVA [108] and Two-tailed [105] test, which are universally known as efficient methods to evaluate the results of statistical models.

As a final comment, our $\mathbf{RQ_4}$ has been designed to follow a focus group methodology, that is, a qualitative research method that, by nature, does not require the participation of a large number of experts. The small dimension of the group—between 5 and 8 members—is explicitly designed to foster discussion and provide insights on the phenomenon of interest. Of course, the reached conclusions might not be definitive and highly generalizable. However, we partially mitigated the first problem because such a qualitative approach is supported alongside quantitative strategies, e.g., data mining and model construction. For the generalization problem, we plan to (1) conduct some replication and (2) conduct surveys with more participants.

6.2 Threats to Internal Validity

Threats to *internal* validity concern factors that might have influenced the obtained results without the researcher's knowledge.

Most of these threats regard the regression models built for the last research question. Because the factors that influence the presence of community smells are a young study field, the possibility of omitting some variables in the model construction is exceptionally high. Therefore, a first mitigation strategy consisted of using all the factors identified by other studies as correlated to community smells. Afterward, we use a mixed-effects model [103, 106] to capture the different time windows of the projects as a random effect, as suggested by the literature. Last, we study the influence of the aforementioned control variables building two baselines models used for evaluating the factors' impact on the response variables.

6.3 Threats to Construct Validity

Threats to construct validity concern with the relation between theory and observation and are mainly due to imprecision in performed measurements.

We can found the first threat in the cultural representations uses in all the present work. In order to characterize culture and cultural dispersion, in all the research questions, we used Hofstede's framework (also known as Hofstede's 6-D model). However, some research works raised serious concerns about this framework [47, 48, 49], even calling for the rejection of its use [50, 51, 52]. One of the most cited critiques is that this model links people's cultural attitudes to their origin country, creating an association of the type "you are born here, so you are in this way." Despite this, Venkateswaran and Ojha, in their systematical analysis of the model and its use [53], contend that Hofstede's framework is the most efficient way to represent the complex world of cultural attitudes.

The second threat regards the sources used. We obtain the results using publicly available datasets previously built:

- the communities dataset from Vasilescu et al. [89];
- the datasets with the community smells by Catolino et al. [69];
- the Hofstede measures obtained from the leading site on the web [12, 44].

The cultural information of the developers and the communities (used in all the research questions) comes from the Vasilescu dataset and the Hofstede one. The process that leads to the identification of the origin countries of the developers performed by Vasilescu, Serebrenik, and Filkov consisted of several steps aimed at refining the results obtained. Of course, we cannot exclude possible imprecisions in the computation of such variables as in the calculation of other control variables from the same dataset. Specifically for the fifth research question, Catolino et al. obtained the number of community smells using the tool CodeFace4Smells that Magnoni [116] and Tamburri et al. [63] evaluated with confident results.

The riskiest threat corresponds to the extremely high number of missed values for the origin country of the developers. In the computation of the cultural metrics, such leakiness could lead to an incorrect cultural characterization of the communities. To avoid such wrong characterization, we used only communities with a few missed countries during our computations.

6.4 Threats to External Validity

Threats to *external* validity are conditions that limit our ability to generalize the results of our experiment to the real world.

As for the generalisability of the results, we have performed a mixed-method approach to capture all the possible related information on the impact of cultural dispersion on community smells. From the quantitative point of view, we have performed a large-scale empirical analysis involving: for the first research questions more than 24 000 open-source projects and 800 000 developers; for the last research question, 22 projects. However, we are aware that the significant presence of missed data for the origin country of the developers leads to the reduction of the usable dataset for our intentions. Therefore, even though the usable data are extensive, further actions to fix such as missed information have to be taken to allow the research to progress.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This chapter illustrates a summary of the research and future prospects.

7.1 Conclusions

This thesis reports on empirical evidence to clarify the connection between *cultural and geographical dispersions* (i.e., how much a development community is different in terms of its members' cultural attitudes and geographical collocation) and problems in communication and collaboration activities. Our intentions can be summarized in the following sentences:

- We want to report information on the cultural and geographical characteristics of GitHub communities.
- We want to report in which way developers consider cultural and geographical differences impact on software development.
- We want to report the impact of cultural and geographical dispersions on the manifestation of community smells.

Intending to be more comprehensive and strengthen research results, we adopted a mixed-method approach, combining qualitative and quantitative research types. The qualitative part consists of a focus group conducted with six open-source developers. The quantitative part consists of a statistical regression model construction to relate the two types of dispersions—cultural and geographical—to four types of community smells, i.e., *Organizational Silo*, *Lone Wolf*, *Black Cloud*, and *Radio Silence*.

The qualitative strategy reveals that developers perceive cultural differences as sources to the manifestation of communication and collaboration problems. Specifically, they consider the combination of individualistic and collectivistic people as one of the most common sources of difficulties. The quantitative strategy reveals that cultural and geographical dispersions impact the emergence of community smells based on their typology. Interestingly, this part of the study also shows a correlation between the presence of individualistic and collectivistic people and the emergence of *Lone Wolf* effects.

Both the strategies lead to very similar results. Then we can conclude, with a reasonable degree of confidence, that cultural and geographical dispersions influence communication and collaboration problems in software communities, bringing to the emergence of community smells. Undoubtedly, further research is needed to deeper investigate such correlations and also to explore new possible additions.

To sum up, our work made the following contributions:

1. We provide a large-scale statistical research on GitHub developers and development communities, which shows their cultural aspects and differences graphically.

2. We provide insights coming from a focus group composed of six participants that expose their opinion on the impact of cultural and geographical dispersions on software development.

3. We provide a statistical regression model that relates the emergence of community smells to the two types of dispersions—cultural and

geographical—of a development community.

4. We provide a GitHub repository [22] to make data, scripts, and other material publicly available.

7.2 Future Work

In this section we describe our future research agenda.

Focus group results generalization. A first step aside to improve our insight is to perform a new qualitative study to enhance the generalizability of the focus group results. In that sense, a survey could be the right tool to use if performed with people with the management of distributed team experience.

Western and eastern cultures. In order to improve our knowledge on the collaborations between people from western and eastern cultures, further research could be performed. For this aim, new datasets are needed cause that the ones used for this study contain very little information on the eastern cultures. Undoubtedly, using more qualitative studies—for example, surveys, interviews, and focus groups—could help.

Private IT company world. The context of our study is influenced by open-source development. The datasets used have been built on GitHub, an open-source platform, and the focus group participants develop open-source software—despite the fact that they are in a private company. For the aforementioned reasons, our results could not be highly accurate for the private IT company world. Further research in the private world is needed in order to expand the generalizability of our results.

Other types of smells. An obvious further work, ables to enhance our findings, regards the types of community smells used for the regression model construction. Specifically, it could be possible to study the correlation between the two types of dispersions—cultural and geographical—and the emergence of other classes of community smells, different from the ones used for this work. However, this task could be challenging to perform cause that the detection strategies for such community smells are not much evolved.

Uncertainty Hiding. In Section 4.4.2, we have proposed a new type of cultural dimension, namely, *Uncertainty Hiding*. This dimension represents the extent to which the members of a culture feel comfortable exposing and revealing risks to ask for help instead of managing them by themselves. Undoubtedly, further research on such attitude is needed to validate the dimension as deserving to exist.

Software Organizational culture. In this study, we have explored the culture of the development communities using their developers as fundamentals "bricks." However, it is well-known that some communities—and also companies—have their specific and established culture [32]. Such a type of culture differs from the countries' ones because they influence the workers' habits during the time spent in the community. A first fascinating study on this could be to see if such cultures can be mapped and represented using a framework like the Hofstede one. Another work regards the way such cultures emerge: it would be interesting to see if the mass mainly influences the culture of the community or if a few "elite" members define it.

ACKNOWLEDGEMENTS

This work is at the end of a long journey that lasted five years and was not without difficulties, as well as beautiful satisfactions. Although I am convinced that actions are worth more than words, I have decided to write down my gratitude addressed to the people who have contributed to the achievement of this personal goal.

I am grateful to Prof. Filomena Ferrucci, my supervisor during this thesis work. During the master's degree, she has always proved to be a competent and skillful figure, as much as willing to help. His continuous commitment to teaching and, in general, to communication are a source of great inspiration for me: a goal I would like to reach one day. I thank her for all that she transmitted to me and for the trust shown repeatedly.

Thanks to Fabio Palomba, one of my co-supervisors during this work. In him, I found a trustworthy, friendly, and competent person capable of supporting the collaborators around him, as any good manager should be able to do. I will do my best to repay all the effort he takes to make the life of his collaborators more serene.

I am grateful to Gemma Catolino, my co-supervisor for this thesis. In her, I have always seen a competent and passionate person, even concerning sensitive issues in the world of research. The compliments received during the most important exam of my career, both towards me and those of the guys I guided, were a source of extreme pride, which I will carry with me for a very long time. I hope to always be worthy of those words, even in the future.

I thank Fabiano Pecorelli, who was my Project Manager during my bachelor's degree. Without him, I would not have taken the path of Software Engineering and Project Management, which makes me really happy. I thank him for the trust he has always shown in me; I hope always to be worthy.

I am grateful to the members of the Software Engineering laboratory of Salerno: Emanuele Iannone, Manuel De Stefano, Valeria Pontillo, Giulia Sellitto, and Giammaria Giordano. Although our time together was not much, I have enjoyed their company immensely over the past few months. Their friendly company and technical skills allowed me to end my journey in the most beautiful way: as in a family.

I enormously thank the guys of the C07 team, who took the Software Engineering exam with me: Gianmario Voria, Viviana Pentangelo, Ciro Maiorino, Giulio Triggiani, Antonio Della Porta, Alessio Casolaro, Nicola Pagliara, and Luca Topo. For them, most of all, there are no words capable of expressing the contribution they have given to my path. Although they do not imagine it, they have helped me in a challenging moment in my life as a man and as a student. They were able to transform an experience that was already very important to me into an even more beautiful memory than I would have imagined. I hope to be able to be with them to enjoy together all the great goals that, without any doubt, they will conquer.

A well-deserved thank goes to Andrea Cupito, my workmate for the entire duration of this journey. His ability to face challenges with skill and a hint of healthy sarcasm has been a pillar for me during all the projects done together. Unfortunately, our paths now separate, but I will never forget all the lessons he gave me and which I will exploit throughout my life.

I am grateful to Alfonso Cavallo, my beloved and irreplaceable cousin. During this last year, he has always been close to me, showing me affection in the most challenging moments. Furthermore, although he does not appreciate my constant talk about work, he has engaged in helping me conduct my research on more than one occasion.

I thank Muriel Rossi that has been one of the most important people in my life. She has always been at my side, and she has supported me for most of my university life, allowing me to improve in both academic and social fields. Although she will probably never read these words, I am grateful to her for changing me for the better.

I thank my family and my extra-academic friends, who have supported me and whom I hope will continue to do so. I will do everything in my power to deserve the help offered to me in the days to come.

In the end, I thank my mother, Blandina. Unfortunately, life was cruel and didn't allow us to spend much time together. Today, my only regret is not being able to show her the man I have become. I thank her because, even today, her memory is a source of enormous inspiration for me: a strength that encourages me to always give my best to achieve my goals. I hope I have become someone she would be proud of.

I conclude these thanks with a consideration dear to me: a thank you is incomplete without a gesture capable of concretely demonstrating one's gratitude. In reflecting on the way to fulfill this, a sentence by an author dear to me came to my mind: *"The most important step a man can take is not the first one, but the next one. Always the next step!"* (Brandon Sanderson, *Oathbringer*). With this in mind, I promise to always commit myself to help others, both in academic and daily life. I intend to repay the support received by giving posterity my commitment, passion, and knowledge. Although limited, I hope they will contribute to building a better and more joyful world for all.

REFERENCES

[1] Narciso Cerpa and June M. Verner. “Why Did Your Project Fail?” In: *Commun. ACM* 52.12 (Dec. 2009), pp. 130–134. ISSN: 0001-0782. DOI: 10.1145/1610252.1610286. URL: https://doi.org/10.1145/1610252.1610286.

[2] Magne Jørgensen and Kjetil Moløkken-Østvold. “How large are software cost overruns? A review of the 1994 CHAOS report”. In: *Information and Software Technology* 48.4 (2006), pp. 297–301.

[3] Jim Johnson. “CHAOS 2020: Beyond Infinity”. In: *Standish Group* (2020).

[4] Taimour Al Neimat. “Why IT projects fail”. In: *The project perfect white paper collection* 8 (2005).

[5] Terry Frieden. “Report: FBI wasted millions on’Virtual Case File’”. In: *Retrieved April* 9 (2005), p. 2005.

[6] Azham Hussain and Emmanuel OC Mkpojiogu. “Requirements: Towards an understanding on why software projects fail”. In: *AIP Conference Proceedings*. Vol. 1761. 1. AIP Publishing LLC. 2016, p. 020046.

[7] Project Management Institute. *A Guide to the Project Management Body of Knowledge*. 7th ed. Aug. 2021, p. 250. ISBN: 1628256648.

[8] Frederick P Brooks Jr. *The mythical man-month: essays on software engineering*. Pearson Education, 1995.

[9] Paul Ralph, Mike Chiasson, and Helen Kelley. "Social Theory for Software Engineering Research". In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. EASE '16. Limerick, Ireland: Association for Computing Machinery, 2016. ISBN: 9781450336918. DOI: 10.1145/2915970.2915998. URL: https://doi.org/10.1145/2915970.2915998.

[10] Sébastien Cherry and Pierre 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.

[11] Sarah Beecham, Padraig OLeary, Ita Richardson, Sean Baker, and John Noll. "Who are we doing global software engineering research for?" In: *2013 IEEE 8th International Conference on Global Software Engineering*. IEEE. 2013, pp. 41–50.

[12] Geert Hofstede, Gert Jan Hofstede, and Michael Minkov. *Cultures and organizations: Software of the mind*. Vol. 2. Mcgraw-hill New York, 2005.

[13] Geert Hofstede. "Dimensionalizing cultures: The Hofstede model in context". In: *Online readings in psychology and culture* 2.1 (2011), pp. 2307–0919.

[14] Damian A Tamburri, Rick Kazman, and Hamed Fahimi. "The architect's role in community shepherding". In: *IEEE Software* 33.6 (2016), pp. 70–79.

[15] Fabio Palomba, Damian Andrew Tamburri, Francesca Arcelli Fontana, Rocco Oliveto, Andy Zaidman, and Alexander Serebrenik. "Beyond Technical Aspects: How Do Community Smells Influence the Intensity

of Code Smells?" In: *IEEE Transactions on Software Engineering* 47.1 (2021), pp. 108–129. DOI: 10.1109/TSE.2018.2883603.

[16] Damian A. Tamburri, Fabio Palomba, and Rick Kazman. "Exploring Community Smells in Open-Source: An Automated Approach". In: *IEEE Transactions on Software Engineering* 47.3 (2021), pp. 630–652. DOI: 10.1109/TSE.2019.2901490.

[17] Damian A. Tamburri. "Software Architecture Social Debt: Managing the Incommunicability Factor". In: *IEEE Transactions on Computational Social Systems* 6.1 (2019), pp. 20–37. DOI: 10.1109/TCSS.2018.2886433.

[18] Sue Wilkinson. "Focus group methodology: a review". In: *International journal of social research methodology* 1.3 (1998), pp. 181–203.

[19] C. Elbert. "Global Software Engineering: Distributed Development, Outsourcing, and Supplier Management." In: *IEEE Computer Society Books* (2010).

[20] Sadhana Deshpande, Ita Richardson, Valentine Casey, and Sarah 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.

[21] Hina Shah, Nancy J. Nersessian, Mary Jean Harrold, and Wendy 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. Bengaluru, India: Association for Computing Machinery, 2012, pp. 77–86. ISBN: 9781450308182. DOI: 10.1145/2160881.2160894. URL: https://doi.org/10.1145/2160881.2160894.

[22] Stefano Lambiase, Fabio Palomba, Gemma Catolino, and Filomena Ferrucci. *Cultural and Geographical Dispersion Impact on Communication and Collaboration of Software Development Teams —*

online appendix. 2021. URL: https://github.com/StefanoLambiase/cultural-dispersion-and-community-smells-online-appendix.

[23] James D Herbsleb and Deependra Moitra. "Global software development". In: *IEEE software* 18.2 (2001), pp. 16–20.

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

[25] Ian Sommerville. *Software Engineering*. 10th ed. Pearson College Div, Aug. 2015, p. 796. ISBN: 0133943038.

[26] Ita Richardson, Valentine Casey, John Burton, and Fergal McCaffery. "Global Software Engineering: A Software Process Approach". In: *Collaborative Software Engineering*. Ed. by Ivan Mistrík, John Grundy, André Hoek, and Jim Whitehead. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35–56. ISBN: 978-3-642-10294-3. DOI: 10.1007/978-3-642-10294-3_2. URL: https://doi.org/10.1007/978-3-642-10294-3_2.

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

[28] Valentine Casey and Ita Richardson. "A structured approach to global software development". In: *European systems and software process improvement and innovation, Dublin, Ireland* (2008).

[29] Sajjad Mahmood, Sajid Anwer, Mahmood Niazi, Mohammad Alshayeb, and Ita Richardson. "Key factors that influence task allocation in global software development". In: *Information and Software Technology* 91 (2017), pp. 102–122. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2017.06.009. URL: https://www.sciencedirect.com/science/article/pii/S0950584917304470.

[30] Viktoria Stray and Nils Brede Moe. "Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack". In: *Journal of Systems and Software* 170 (2020), p. 110717. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2020.110717. URL: https://www.sciencedirect.com/science/article/pii/S0164121220301564.

[31] R. Kreitner, A. Kinicki, and M. Buelens. *Organizational Behavior*. First European Edition ed. London UK. McGraw-Hill Publishing Company, 1999.

[32] Charles Handy. *Understanding Organizations*. 4th ed. Penguin, Feb. 1993, p. 448. ISBN: 0140156038.

[33] Sandra Dawson. *Analysing organisations*. Macmillan International Higher Education, 1996.

[34] Srinivas Krishna, Sundeep Sahay, and Geoff Walsham. "Managing cross-cultural issues in global software outsourcing". In: *Communications of the ACM* 47.4 (2004), pp. 62–66.

[35] Judith S Olson and Gary 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." In: *Queue* 1.9 (2003), pp. 52–59.

[36] Jessica Lipnack and Jeffrey Stamps. *Virtual teams: People working across boundaries with technology*. John Wiley & Sons, 2008.

[37] Anne F Rutkowski, Douglas R Vogel, Michiel Van Genuchten, Theo MA Bemelmans, and Marc Favier. "E-collaboration: The reality of virtuality". In: *IEEE Transactions on professional communication* 45.4 (2002), pp. 219–230.

[38] Dale Walter Karolak. *Global software development: managing virtual teams and environments*. IEEE Computer society press, 1999.

[39] Gul Calikli, Ayse Bener, and Berna 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.

[40] Valentine Casey. "Imparting the importance of culture to global software development". In: *ACM inroads* 1.3 (2011), pp. 51–57.

[41] Adrian Furnham. *The psychology of behaviour at work: The individual in the organization*. Psychology press, 2012.

[42] Edward T Hall. "The silent language in overseas business". In: *Harvard business review* 38.3 (1960), pp. 87–96.

[43] Charles Hampden-Turner, Fons Trompenaars, and Charles Hampden-Turner. *Riding the waves of culture: Understanding diversity in global business*. Hachette UK, 2020.

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

[45] Greg 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.

[46] Sameer Abufardeh and Kenneth 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.

[47] Karlene H Roberts and Nakiye 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] Galit Ailon. “Mirror, mirror on the wall: Culture’s consequences in a value test of its own design”. In: *Academy of management review* 33.4 (2008), pp. 885–904.

[49] Rachel F Baskerville. “Hofstede never studied culture”. In: *Accounting, organizations and society* 28.1 (2003), pp. 1–14.

[50] Paul Brewer and Sunil Venaik. “On the misuse of national culture dimensions”. In: *International Marketing Review* (2012).

[51] Paul Brewer and Sunil Venaik. “The ecological fallacy in national culture research”. In: *Organization Studies* 35.7 (2014), pp. 1063–1086.

[52] Arndt Sorge. In: *Administrative Science Quarterly* 28.4 (1983), pp. 625–629. ISSN: 00018392. URL: `http://www.jstor.org/stable/2393017`.

[53] Ramya T Venkateswaran and Abhoy K Ojha. “Abandon Hofstede-based research? Not yet! A perspective from the philosophy of the social sciences”. In: *Asia Pacific Business Review* 25.3 (2019), pp. 413–434.

[54] Fabio Palomba, Gabriele Bavota, Massimiliano Di Penta, Fausto Fasano, Rocco Oliveto, and Andrea De Lucia. “On the diffuseness and the impact on maintainability of code smells: a large scale empirical investigation”. In: *Empirical Software Engineering* 23.3 (2018), pp. 1188–1221.

[55] Michele Tufano, Fabio Palomba, Gabriele Bavota, Rocco Oliveto, Massimiliano Di Penta, Andrea De Lucia, and Denys Poshyvanyk. “When and why your code starts to smell bad (and whether the smells go away)”. In: *IEEE Transactions on Software Engineering* 43.11 (2017), pp. 1063–1088.

[56] Naouel Moha, Yann-Gaël Guéhéneuc, Laurence Duchien, and Anne-Francoise Le Meur. “Decor: A method for the specification and detection of code and design smells”. In: *IEEE Transactions on Software Engineering* 36.1 (2009), pp. 20–36.

[57] Fabio Palomba, Annibale Panichella, Andrea De Lucia, Rocco Oliveto, and Andy Zaidman. "A textual-based technique for smell detection". In: *2016 IEEE 24th international conference on program comprehension (ICPC)*. IEEE. 2016, pp. 1–10.

[58] Stefano Lambiase, Andrea Cupito, Fabiano Pecorelli, Andrea De Lucia, and Fabio Palomba. "Just-In-Time Test Smell Detection and Refactoring: The DARTS Project". In: *Proceedings of the 28th International Conference on Program Comprehension*. 2020, pp. 441–445.

[59] Ward Cunningham. "The WyCash portfolio management system". In: *ACM SIGPLAN OOPS Messenger* 4.2 (1992), pp. 29–30.

[60] Melvin E Conway. "How do committees invent". In: *Datamation* 14.4 (1968), pp. 28–31.

[61] D.A. Tamburri, P. Kruchten, P. Lago, and H. van Vliet. "Social Debt in Software Engineering: Insights from Industry". English. In: *Journal of Internet Services and Applications* (2015). ISSN: 1867-4828. DOI: 10.1186/s13174-015-0024-6.

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

[63] Damian Andrew Andrew Tamburri, Fabio Palomba, and Rick Kazman. "Exploring community smells in open-source: An automated approach". In: *IEEE Transactions on software Engineering* (2019).

[64] Mitchell Joblin, Wolfgang Mauerer, Sven Apel, Janet Siegmund, and Dirk Riehle. "From developer networks to verified communities: A fine-grained approach". In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. Vol. 1. IEEE. 2015, pp. 563–573.

[65] Michele Tufano, Fabio Palomba, Gabriele Bavota, Rocco Oliveto, Massimiliano Di Penta, Andrea De Lucia, and Denys Poshyvanyk. “When and why your code starts to smell bad”. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. Vol. 1. IEEE. 2015, pp. 403–414.

[66] Nuri Almarimi, Ali Ouni, and Mohamed Wiem Mkaouer. “Learning to detect community smells in open source software projects”. In: *Knowledge-Based Systems* 204 (2020), p. 106201.

[67] Nuri Almarimi, Ali Ouni, Moataz Chouchen, Islem Saidani, and Mohamed Wiem Mkaouer. “On the detection of community smells using genetic programming-based ensemble classifier chain”. In: *Proceedings of the 15th International Conference on Global Software Engineering*. 2020, pp. 43–54.

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

[69] Gemma Catolino, Fabio Palomba, Damian A Tamburri, Alexander Serebrenik, and Filomena 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.

[70] Gemma Catolino, Fabio Palomba, Damian A Tamburri, Alexander Serebrenik, and Filomena 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.

[71] Antonio Martini and Jan Bosch. “Revealing social debt with the CAFFEA framework: An antidote to architectural debt”. In: *2017*

IEEE International Conference on Software Architecture Workshops (ICSAW). IEEE. 2017, pp. 179–181.

[72] Damian A Tamburri, Fabio Palomba, Alexander Serebrenik, and Andy Zaidman. "Discovering community patterns in open-source: a systematic approach and its evaluation". In: *Empirical Software Engineering* 24.3 (2019), pp. 1369–1417.

[73] Guilherme Avelino, Leonardo Passos, Andre Hora, and Marco Tulio Valente. "A novel approach for estimating truck factors". In: *2016 IEEE 24th International Conference on Program Comprehension (ICPC)*. IEEE. 2016, pp. 1–10.

[74] Damian A Tamburri and Elisabetta Di Nitto. "When software architecture leads to social debt". In: *2015 12th Working IEEE/IFIP Conference on Software Architecture*. IEEE. 2015, pp. 61–64.

[75] Helen Sharp and Hugh Robinson. "Some social factors of software engineering: the maverick, community and technical practices". In: *Proceedings of the 2005 workshop on Human and social factors of software engineering*. 2005, pp. 1–6.

[76] S Magnoni, DA Tamburri, E Di Nitto, and R Kazman. "Analyzing quality models for software communities". In: *Communications of the ACM-: Under Review* (2017).

[77] Marcelo Cataldo, James D Herbsleb, and Kathleen M Carley. "Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity". In: *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. 2008, pp. 2–11.

[78] Marcelo Cataldo, Patrick A Wagstrom, James D Herbsleb, and Kathleen 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.

[79] Giuseppe Valetto, Sunita Chulani, and Clay Williams. "Balancing the value and risk of socio-technical congruence". In: *Workshop on Sociotechnical Congruence*. Vol. 4. Citeseer. 2008.

[80] Melvin E Conway. "How do committees invent". In: *Datamation* 14.4 (1968), pp. 28–31.

[81] Irwin Kwan, Adrian Schroter, and Daniela Damian. "Does socio-technical congruence have an effect on software build success? a study of coordination in a software project". In: *IEEE Transactions on Software Engineering* 37.3 (2011), pp. 307–324.

[82] Erik Trainer, Stephen Quirk, Cleidson de Souza, and David Redmiles. "Bridging the gap between technical and social dependencies with ariadne". In: *Proceedings of the 2005 OOPSLA workshop on Eclipse technology eXchange*. 2005, pp. 26–30.

[83] Anita Sarma, Larry Maccherone, Patrick Wagstrom, and James Herbsleb. "Tesseract: Interactive visual exploration of socio-technical relationships in software development". In: *2009 IEEE 31st International Conference on Software Engineering*. IEEE. 2009, pp. 23–33.

[84] Fabio Palomba, Damian A Tamburri, Alexander Serebrenik, Andy Zaidman, Francesca Arcelli Fontana, and Rocco Oliveto. "Poster: How Do Community Smells Influence Code Smells?" In: *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*. IEEE. 2018, pp. 240–241.

[85] John W Creswell and J David Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[86] Massimiliano Di Penta and Damian Andrew Tamburri. "Combining quantitative and qualitative studies in empirical software engineering research". In: *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE. 2017, pp. 499–500.

[87] R Burke Johnson and Anthony J Onwuegbuzie. "Mixed methods research: A research paradigm whose time has come". In: *Educational researcher* 33.7 (2004), pp. 14–26.

[88] Juliet Corbin and Anselm Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory.* Sage publications, 2014.

[89] Bogdan Vasilescu, Alexander Serebrenik, and Vladimir 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.

[90] Michelle Cleary, Jan Horsfall, and Mark Hayter. "Data collection and sampling in qualitative research: does size matter?" In: *Journal of advanced nursing* (2014), pp. 473–475.

[91] Orlandrew E Danzell and Lisandra M Maisonet Montañez. "Understanding the lone wolf terror phenomena: assessing current profiles". In: *Behavioral Sciences of Terrorism and Political Aggression* 8.2 (2016), pp. 135–159.

[92] Stanley Wasserman, Katherine Faust, et al. "Social network analysis: Methods and applications". In: (1994).

[93] Fabio Palomba, Mario Linares-Vásquez, Gabriele Bavota, Rocco Oliveto, Massimiliano Di Penta, Denys Poshyvanyk, and Andrea De Lucia. "Crowdsourcing user reviews to support the evolution of mobile apps". In: *Journal of Systems and Software* 137 (2018), pp. 143–162.

[94] Norman Fenton and James Bieman. *Software metrics: a rigorous and practical approach.* CRC press, 2019.

[95] Peter H Rossi, James D Wright, and Andy B Anderson. *Handbook of survey research*. Academic Press, 2013.

[96] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark G.J. van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov. "Gender and Tenure Diversity in GitHub Teams". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2015, pp. 3789–3798. ISBN: 9781450331456. URL: `https://doi.org/10.1145/2702123.2702549`.

[97] Sarah MacCurtain, Patrick C Flood, Nagarajan Ramamoorthy, Michael A West, and Jeremy F Dawson. "The top management team, reflexivity, knowledge sharing and new product performance: A study of the Irish software industry". In: *Creativity and Innovation Management* 19.3 (2010), pp. 219–232.

[98] Peter Michael Blau. *Inequality and heterogeneity: A primitive theory of social structure*. Vol. 7. Free Press New York, 1977.

[99] Giuseppe Valetto, Mary Helander, Kate Ehrlich, Sunita Chulani, Mark Wegman, and Clay 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.

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

[101] Mívian Ferreira, Marco Tulio Valente, and Kecia 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.

[102] John-Paul Hatala and Joseph George Lutta. "Managing information sharing within an organizational setting: A social network perspective". In: *Performance Improvement Quarterly* 21.4 (2009), pp. 5–33.

[103] Mary J Lindstrom and Douglas M Bates. "Newton—Raphson and EM algorithms for linear mixed-effects models for repeated-measures data". In: *Journal of the American Statistical Association* 83.404 (1988), pp. 1014–1022.

[104] Skipper Seabold and Josef Perktold. "Statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[105] David B Pillemer. "One-versus two-tailed hypothesis tests in contemporary educational research". In: *Educational Researcher* 20.9 (1991), pp. 13–17.

[106] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. "Fitting linear mixed-effects models using lme4". In: *arXiv preprint arXiv:1406.5823* (2014).

[107] Robert M O'brien. "A caution regarding rules of thumb for variance inflation factors". In: *Quality & quantity* 41.5 (2007), pp. 673–690.

[108] Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. "An anova test for functional data". In: *Computational statistics & data analysis* 47.1 (2004), pp. 111–122.

[109] Daniel Standage. *Literate programming, RStudio, and ipython notebook*. Mar. 2015. URL: `https://biowize.wordpress.com/2015/03/13/literate-programming-rstudio-and-ipython-notebook/`.

[110] Ayushi Rastogi, Nachiappan Nagappan, Georgios Gousios, and André van der Hoek. "Relationship between geographical location and evaluation of developer contributions in github". In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2018, pp. 1–8.

[111] Reza Nadri, Gema Rodriguezperez, and Meiyappan Nagappan. "On the Relationship Between the Developer's Perceptible Race and Ethnicity

and the Evaluation of Contributions in OSS". In: *IEEE Transactions on Software Engineering* (2021).

[112] Young Hoon Kwak and Kenneth Scott LaPlace. "Examining risk tolerance in project-driven organization". In: *Technovation* 25.6 (2005), pp. 691–695.

[113] Kenneth P Burnham and David R Anderson. "Multimodel inference: understanding AIC and BIC in model selection". In: *Sociological methods & research* 33.2 (2004), pp. 261–304.

[114] Hirotogu Akaike. "Information theory and an extension of the maximum likelihood principle". In: *Selected papers of hirotugu akaike*. Springer, 1998, pp. 199–213.

[115] Dirk Homscheid and Mario Schaarschmidt. "Between organization and community: investigating turnover intention factors of firm-sponsored open source software developers". In: *Proceedings of the 8th ACM Conference on Web Science*. 2016, pp. 336–337.

[116] Simone Magnoni. "An approach to measure community smells in software development communities". MA thesis. Politecnico di Milano - Scuola di Ingegneria Industriale e dell'Informazione, July 2016.

LIST OF FIGURES

LIST OF TABLES