Cultural and Socio-Technical Aspects in Software Development

Stefano Lambiase
slambiase@unisa.it
SeSa Lab, Department of Computer Science, University of Salerno
Fisciano (SA), Italy

ABSTRACT

Software development is essentially a collaborative, socio-technical endeavor where the interplay between stakeholders and technical elements is integral. This synergy becomes particularly crucial in the context of geographically dispersed teams, a practice that is becoming more prevalent. Despite the ubiquity of this nature, the current body of research in Global Software Development (GSD) encounters limitations, rendering the attained results less accessible for practical implementation by industry professionals. Moreover, the role of social debt, the additional cost derived by adopting socio-technical anti-patterns, in GSD still needs to be deepened. This Ph.D. research project aims to surmount these challenges by constructing a robust theoretical foundation for effectively managing socio-technical aspects—particularly in the form of factors related to social debt—in software development, with a keen focus on their correlation with cultural differences within software teams. The framework systematically captures and examines cultural differences, investigating their ramifications on various facets of software development while exploring practical strategies employed by practitioners to navigate these influences. Furthermore, the project aspires to make substantial contributions to the professional software development realm by translating research findings into tangible tools for practitioners. This framework is designed not only for immediate application but also to facilitate project success through heightened cultural awareness and adaptability. Ultimately, it strives to enhance the well-being of developers working in inclusive and culturally diverse environments.

CCS CONCEPTS

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

KEYWORDS

Global Software Development; Cultural Dispersion; Socio-Technical Aspects; Social Debt; Community Smells

ACM Reference Format:
Stefano Lambiase. 2024. Cultural and Socio-Technical Aspects in Software Development. In *28th International Conference on Evaluation and Assessment in Software Engineering (EASE 2024), June 18–21, 2024, Salerno, Italy.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3661167.3661230

ADVISOR INFORMATION

Name Filomena Ferrucci
E-mail fferrucci@unisa.it
Affilition University of Salerno, Italy

1 MOTIVATION, RATIONALE, AND CONTRIBUTION OF THE RESEARCH

1.1 Social Debt and Socio-Technical Aspects

Software development and engineering inherently function as social activities [6, 20, 31] that bring together organizations, managers, developers, and stakeholders from diverse global locations for collective pursuits [9, 15, 32]. This cooperative dynamic introduces challenges for both engineers and managers, especially in the domains of collaboration and communication, giving rise to issues like personality conflicts, language barriers, and cultural differences. Furthermore, the intricate interplay between the social aspects of software development and its technological components, encompassing both the software product and the tools utilized during work, has prompted the research community to shift its focus towards socio-technical aspects rather than purely social ones. This redefinition is crucial; it underscores the necessity for all studies exploring collaboration in the software development context to consistently factor in the role of technology as a control variable.

Investigating socio-technical aspects, prior research has delved into the concept of *social debt* (inspired by the well-known concept of technical debt), which refers to unforeseen project costs linked to suboptimal development community dynamics [37, 38]. Additionally, the focus has extended to phenomena known as *community smells* [36], encompassing socio-technical characteristics (e.g., high formality) and patterns (e.g., recurrent condescending behavior or rage-quitting). These elements can contribute to the emergence of social debt [29, 35, 36, 39]. Recently, the research community has explored the diffusion and impact of community smells, along with factors correlated with their emergence [3, 4, 30].

Although the spread of research on socio-technical aspects has increased over the years, some limitations still need to be addressed.

Lack in Using Catalogues. Many studies tend to refrain from using the concepts developed by research in order to make the results on socio-technical aspects more measurable and applicable (e.g., Community Smells) [7]. This choice leads to an inevitable fragmentation of knowledge, as well as community, and a resulting weakening of the relevance of research on the topic.

Lack of Tools. To date, the number of tools designed for practitioners and capable of supporting them in managing socio-technical aspects still needs to be increased. Further effort in this regard is needed in order to build strong links between the worlds of research and practitioners.

1.2 Global Software Development and Culture

Within the challenges potentially affecting software development, the concept of *culture* is emerging as a pivotal element requiring thorough consideration from members of the software community across the entire development lifecycle. *Culture* is delineated as shared motives, values, beliefs, identities, and interpretations or meanings of significant events derived from common experiences among members of collectives, transmitting across generations [22].

The study of culture in software development—which pertains to the Global Software Development community—demonstrated that culture can impact various aspects of software development in a manner that extends beyond observable and measurable dimensions. Despite acknowledging the considerable effort invested in constructing a substantial body of knowledge on culture in software engineering, our research identifies several fundamental limitations in the current state of the art:

Lack of Studies on Socio-Technical Aspects. Prior research has examined the interplay between culture and software development by focusing on specific processes and product metrics. For instance, certain studies have delved into unraveling the influence of culture on the code review process [5] or code quality [1]. Despite these investigations, there is still a lack of knowledge concerning how culture might shape the behavioral patterns among developers [44], particularly in terms of socio-technical aspects (e.g., community smells). As a result, *our comprehension of the impact of culture on socio-technical dimensions remains limited.*

Lack in Using Cultural Frameworks. It is worth noting that a limited number of studies have employed cultural frameworks [13, 14, 18, 19], which offer (1) a set of cultural behaviors associated with individuals and (2) numerical values to characterize these behaviors. Furthermore, previous research has predominantly regarded culture as an "abstract" concept without quantifying or characterizing it [27]. Due to this approach, most of the findings in the literature cannot be assessed against a reference framework, *thus impeding a more objective understanding of how culture influences software engineering practices.*

Lack of Theoretical Framework. No comprehensive theory sufficiently clarifies the influence of cultural differences on software development and provides effective strategies for handling them. This absence holds two noteworthy implications: firstly, *it contributes to fragmented research*, often resulting in new contributions without substantial progress in the field; secondly, *it limits the practical applicability of research findings in the software development industry*, as there is no established framework readily understandable and implementable by practitioners.

1.3 Research Objective and Contributions

Building on the limitations mentioned above, this work focused on studying socio-technical aspects and their relationship with cultural behaviors. The final aim is to develop a theoretical framework intended to encompass how these two aspects relate and explore strategies employed by practitioners to (1) manage potentially related issues and (2) benefit from such heterogeneity. The guiding hypothesis posits that cultural differences significantly influence the development lifecycle and its participants, potentially influencing them positively and negatively depending on how well such "dispersion" is managed.

In addition to the goal above, this work aims to contribute substantially to the practitioners' software development landscape. To achieve this, achieved research findings are intended to be transferred into actionable results by creating tools that embed the new knowledge. Such tools are planned to be informed by the research and developed according to guidelines from the practice to make them usable and appealing to practitioners.

Practically, this research contributes to the state of the art in software engineering by means of the following contributions:

(1) First, an analysis of the literature on socio-technical aspects and social debt is intended to be carried out, and consequential knowledge on the matter is planned to be provided in the form of research papers;
(2) Second, from the various aspects related to socio-technical aspects, culture and its heterogeneity in development teams is intended to be put as protagonists of ulterior research, consequentially providing knowledge on such relationships;
(3) Last, more as a transversal objective, a set of tools are planned to be developed in order to make the results achieved during the investigations usable by practitioners.

2 RESEARCH QUESTIONS

Concerning the first objective, three research questions have been formulated to guide the study [42]. No research question have been formulated for the second objective since it is a technology transfer process rather than a research one.

RQ$_1$—*What is the current state of the art regarding socio-technical aspects and cultural aspects in software development, and what are the associated limitations in the existing research?*

The primary objective of the initial inquiry is to offer an overview of the current state of the art in the main subject under analysis and identify possible gaps that could impact the key findings of the dissertation. Additionally, the examination of the state of the art is not solely intended to generate new insights but also to establish the foundation for original outcomes. Such a research question is divided into two sub-questions: the first one is oriented on socio-technical aspects, while the second one is focused on cultural differences.

RQ$_2$—*How do cultural differences in software development teams influence socio-technical aspects?*

The focal point of the Ph.D. project resides in the second research question, embodying its primary purpose. It is expected that numerous more specific research queries is planned to arise both from this question and the responses to the first question, each evolving into distinct endeavors. The synthesis of these endeavors is anticipated to result in a comprehensive response to the initial question. Moreover, cultural aspects are not the only focus of this process; to capture the overall phenomena, an analysis including potential control factors is intended to be carried out to form the theoretical background of the work principal investigator.

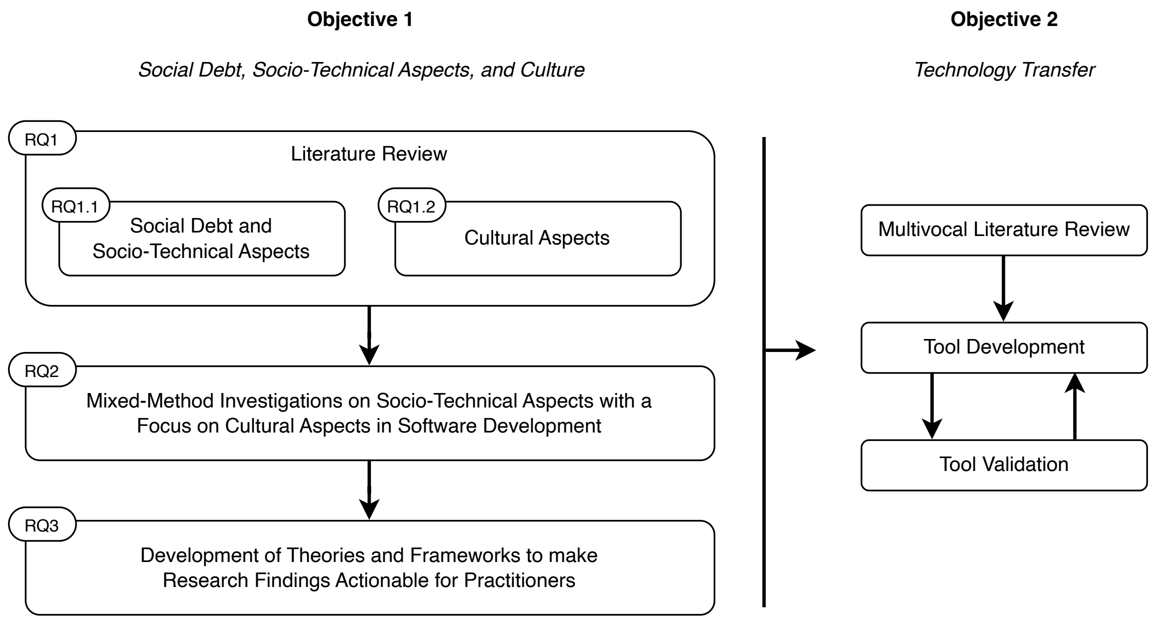
Objective 1
Social Debt, Socio-Technical Aspects, and Culture
RQ1
Literature Review
RQ1.1
Social Debt and Socio-Technical Aspects
RQ1.2
Cultural Aspects
RQ2
Mixed-Method Investigations on Socio-Technical Aspects with a Focus on Cultural Aspects in Software Development
RQ3
Development of Theories and Frameworks to make Research Findings Actionable for Practitioners
Objective 2
Technology Transfer
Multivocal Literature Review
Tool Development
Tool Validation

Figure 1: Research Activity Process.

RQ₃—*Which theoretical frameworks could be used to represent and make the findings of the research actionable?*

During the second phase, as the Ph.D. period approaches its conclusion, the knowledge acquired will be applied to conduct studies with the aim of developing comprehensive theoretical frameworks, representing the primary contribution of the dissertation. By doing so, actionable knowledge will arise that will consequentially support both practitioners and researchers. The former will be able to use and absorb the knowledge, while the latter will also be capable of extending the frameworks.

3 WORK PLAN

To tackle the initial research question, a comprehensive literature review is intended to be carried out to examine credible sources extensively. The objective is to identify secondary studies, such as systematic literature reviews and mapping studies, related to the topic. If secondary studies are not discerned, the first step of the research process involves conducting a systematic literature review and a mapping study.

Regarding the second research question, the strategy involves partitioning the research into two distinct phases. In the initial phase, *mixed-method research approaches* [10] will be employed to delve into socio-technical and cultural aspects in contexts lacking prior contributions highlighted during the process for the first research question. This research methodology incorporates both qualitative methods (such as qualitative analysis, grounded theory, and interviews) and quantitative investigations (including statistically supported empirical studies and data mining) conducted on the same dataset or within the same context to explore similar research inquiries. The ultimate objective is to attain *theoretical saturation*, denoting the stage in category development where no new properties, dimensions, or relationships emerge during analysis. Hence, the theory is considered saturated when both qualitative and quantitative data converge on identical conclusions. Conversely, any disparities between the two studies would necessitate further investigations into the matter.

Concerning the final research question, a combination of qualitative and quantitative approaches will once again be utilized. The development of theories from qualitative data will be guided by the *Grounded Theory* [16, 34] approach. Specifically, a socio-technical grounded theory for data analysis [16] will be applied, seamlessly aligning with the context under analysis. On the quantitative front, *structural equation modeling* [33] will be employed to establish a theory supported by quantitative data. Ultimately, these two types of theories will be collectively assessed to formulate a unified theory supported by both quantitative and qualitative data.

Regarding the step focused on technology transfer, first, a multivocal literature review [12] will be conducted to collect insights from both researchers and practitioners (the primary target audience of this part of the project). Once helpful information on the tools to be developed is collected, development will begin through an iterative approach designed to improve the quality of the final product over time.

Figure 1 reports the research method defined to answer the research questions and address the research objectives. Moreover, we plan to use the *ACM/SIGSOFT Empirical Standards* for all the research associated to the project.

3.1 Metrics and Variables Definition

3.1.1 Socio-Technical Aspects—Community Smells. As mentioned previously, socio-technical aspect and their relations with social debt, represents the context of the investigation and our ultimate dependent variable. For such a reason, we decided to focus our attention on *Community Smells*, i.e., sub-optimal patterns across

Table 1: 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.
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.
Disengagement	Thinking the product is mature enough and sending it to operations even though it might not be ready.

the organizational and social structure in a software development community that are precursors of alarming and unforeseen socio-technical events [35, 38]. Table 1 reports the definition of some of the most used smells in literature.

3.1.2 Cultural Dispersion. Regarding the central theme of our discussions, it is crucial to recognize the inherent complexities when addressing culture in our investigations and the potential for misunderstandings. We have taken deliberate steps to ensure clarity, alignment, and to mitigate the risk of inaccurate results.

Specifically, we have chosen to represent culture using cultural frameworks [18, 19] rather than relying solely on the definition of culture, which often introduces interpretation issues. These frameworks portray culture through a set of "dimensions," each emphasizing differences in people's behaviors and employing numerical values to measure the extent of these differences. For example, Hofstede's *Uncertainty Avoidance* dimension assesses a society's comfort with uncertainty [18]. High scores indicate a preference for rules and stability, avoiding ambiguity, while low scores suggest openness to change, risk-taking, and adaptability to uncertainty.

Expanding on this foundation, we have introduced the concept of *cultural dispersion* as a metric (quantitative) or representation (qualitative) of how much a community varies in terms of its members' cultural background and behavior. This operationalization of cultural frameworks is twofold: qualitatively, we represent culture using the described behaviors; quantitatively, we rely on the values associated with the dimensions. By approaching cultural aspects through empirically linked behaviors, we ensure a precise analysis, eliminating the interpretative nature associated with the definition of culture [18, 19]. This method allows us to delve into cultural dimensions through tangible behaviors established by research, ensuring clarity and minimizing subjective interpretation.

3.2 Data Collection Approaches

Both qualitative and quantitative studies will be carried out regarding data collection. Related to quantitative studies, data from repositories like GitHub and similar will be collected on needed and used. For such a purpose, data mining tools will be operationalized. Regarding qualitative studies, interviews and similar will be used; transcripts of sessions will be collected if needed. Moreover, information on participants will be used when necessary. For both qualitative and quantitative studies, questionnaires will be used.

All the data collection process will follow rigid rules regarding ethical and privacy concern. Moreover, each study involving human will receive approval from an ethical board before conducted.

3.3 Risk Management Strategies

The following section reports a brief outline of the potential risks affecting the project. First of all, cultural aspects are not easy to capture and study; the risk of falling into unconcluded results or even not obtaining any is concrete. For such a reason, a mixed-method approach is planned, aiming at maximizing the number of sources for findings. Moreover, collaborations with individuals from different fields, as the one of cross-cultural business management, are intended to be done. Regarding the risk of not obtaining data, this is particularly related to the qualitative part of the work. Concerning survey studies (which involve administering questionnaires), platforms for facilitating them are intended to be used. Regarding interviews, no similar instruments are available. In the worst case, the research will rely heavily on qualitative surveys. Last but not least, culture and its implications are controversial by nature and raise profound ethical considerations. Ethical boards are planned to be involved in the process to ensure a good research process.

4 ACHIEVED RESULTS

In terms of results, they are discussed in alignment with the objectives mentioned in previous sections.

4.1 Main Research Results

Addressing the first objective, the literature analysis identified limitations, which were detailed in the paper's introduction, guiding subsequent steps and answering the first research question. Regarding the study of literature on socio-technical aspects, we found an already and recently published systematic literature review on the matter [7]. Concerning the study of cultural aspects, we performed a literature review on our own.

Concerning the second research question, the initial focus was on exploring how cultural differences impact the emergence of community smells and the productivity of a development community. Culture, treated through the concept of *cultural dispersion*, was systematically approached to provide a concrete understanding that led to three publications [24–26]:

- The exploration of how culture impacts the social aspects of software development shaped the foundation of the initial study. Our hypothesis is that cultural dispersion might influence collaboration, consequently giving rise to community smells. The study revealed that cultural dispersion indeed has an impact on the emergence of all community smells, with nuanced outcomes that indicate not solely negative effects. One noteworthy

discovery was the correlation between the presence of individualistic and collectivist individuals and the emergence of Lone Wolf effects. The findings from this work [26] were published at the *International Conference on Software Engineering-Software Engineering in Society* (ICSE-SEIS 2022).
- Broadening our examination of socio-technical metrics, we delved into the connections between cultural dispersion and productivity. Through a mixed-method study, we uncovered that dispersion metrics can exert both positive and negative influences on productivity, contingent upon how managers address cultural differences. For instance, the integration of individualistic and collaboration-oriented individuals might prove ineffective, highlighting the necessity for a nuanced approach. This research led to the publication of two papers [24, 25]: the first [24] at the *Software Engineering and Advanced Applications Euromicro Conference* (SEAA 2022), and the second [25] (an extension of the first) in the *Journal of Systems and Software* (JSS).

Moreover, in each study, we aimed to evaluate our representation of culture through cultural dispersion metrics and concepts, confirming its maturity for practical use by practitioners and researchers.

Related to the last research question, Socio-Technical Grounded Theory was used to investigate the relationship between cultural dispersion and software development. This work, as a sum of all the knowledge gathered so far, resulted in a preliminary theory composed of 6 theoretical models. In this theories, we delve into the challenges and consequences posed by cultural variations within these teams while also proposing strategies for addressing potential issues. To enhance its practicality, we drew inspiration from the GLOBE framework of culture, thus centering it on the category of **"Dealing With Cultural Dispersion"**. The work [23] has been accepted at the *International Conference on Software Engineering-Software Engineering in Society* (ICSE-SEIS 2024).

4.2 Technology Transfer

Concerning the second objective, we identified conversational agents as a managerial tool in the context of software development. Initially, we conducted a comprehensive literature review (currently under revision) to gather insights into the adoption and challenges associated with these tools in the software engineering domain.

Building upon the acquired knowledge, we introduced CADOCS (**C**onversational **A**gent for the **D**etection **O**f **C**ommunity **S**mells) [41], a conversational agent presented at the *International Conference on Software Maintenance and Evolution* (ICSME 2022). CADOCS extends a previous community smell detection tool proposed by Almarimi et al. [2], aiming to (1) improve its usability and (2) enhance its functionality by providing initial support for software analytics instruments crucial for diagnosing and refactoring community smells. Additionally, CADOCS has been designed for high extensibility, ensuring continuous improvement over time and the seamless integration of emerging insights from the research.

5 RELATED WORK

5.1 Socio-Technical Aspects

In the field of Software Engineering (SE), the concept of community smells [36] has garnered attention, highlighting sub-optimal patterns within the organizational structures of software development communities. These patterns are seen as early indicators of potential socio-technical issues that could lead to social debt [38].

Research has demonstrated the significant impact of community smells on software quality. For instance, Palomba et al. [29] found that community smells are a leading cause of code smells, which are suboptimal coding practices that can lead to software defects or failures. Additionally, studies have explored how community smells affect other aspects of software engineering, such as architectural debt and organizational structures, underscoring their prevalence in open-source teams and their perceived importance by developers [28, 39, 40].

Building on these insights, Palomba and Tamburri [30] introduced a machine learning model to predict community smells based on socio-technical metrics, achieving an F-Measure of 78%. Similarly, Almarimi et al. [4] developed a genetic algorithm-based model to identify eight common community smells across 103 open-source projects, showing superior performance with an F-measure of 89%.

In terms of addressing and mitigating community smells, Catolino et al. [8] investigated the influence of socio-technical factors on the variability of these smells and the strategies developers employ to eliminate them. This research has led to the identification of specific refactoring practices for addressing community smells detected by CodeFace4Smells.

5.2 Cultural Aspects in Software Egnineering

Researchers in Global Software Engineering have examined aspects related to distributed software development. Culture has emerged as a pivotal factor, prompting researchers to delve into its impact throughout the entire lifecycle [5, 21, 43].

One noteworthy contribution comes from Borchers [5], who explored the influence of cultural factors on software engineering processes, such as code review, with a specific focus on three distinct countries: Japan, India, and the United States. This study operationalized the Hofstede cultural framework [17] and highlighted how different cultures approached software engineering practices uniquely.

Another notable contribution comes from Yasin et al. [43]. They conducted an empirical investigation involving experimentation and surveys to determine how group activities can mitigate the emergence of culturally-originated problems. Their discussion sessions and survey results demonstrated their ability to identify critical GSE challenges, especially those related to teamwork, in a simulated scenario.

Moreover, Deshpande et al. [11] conducted a series of qualitative studies (i.e., questionnaire, telephonic interviews, and structured interviews) with 15 project managers to study the way they face challenges derived by cultural differences in software development Indian teams. Consequentially, the authors tried to provide a list of practical strategies that practitioners can use to deal with culturally originated problems.

REFERENCES

[1] Sameer Abufardeh and Kenneth Magel. 2010. The impact of global software cultural and linguistic aspects on Global Software Development process (GSD): Issues and challenges. In *4th International conference on new trends in information science and service science*. IEEE, 133–138.

[2] Nuri Almarimi, Ali Ouni, Moataz Chouchen, and Mohamed Wiem Mkaouer. 2021. csDetector: an open source tool for community smells detection. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1560–1564.
[3] Nuri Almarimi, Ali Ouni, Moataz Chouchen, Islem Saidani, and Mohamed Wiem Mkaouer. 2020. On the detection of community smells using genetic programming-based ensemble classifier chain. In *Proceedings of the 15th International Conference on Global Software Engineering*. 43–54.
[4] Nuri Almarimi, Ali Ouni, and Mohamed Wiem Mkaouer. 2020. Learning to detect community smells in open source software projects. *Knowledge-Based Systems* 204 (2020), 106201.
[5] Greg Borchers. 2003. The software engineering impacts of cultural factors on multi-cultural software development teams. In *25th International Conference on Software Engineering, 2003. Proceedings.* IEEE, 540–545.
[6] Frederick P Brooks Jr. 1995. *The mythical man-month: essays on software engineering.* Pearson Education.
[7] Eduardo Caballero-Espinosa, Jeffrey C Carver, and Kimberly Stowers. 2023. Community smells—The sources of social debt: A systematic literature review. *Information and Software Technology* 153 (2023), 107078.
[8] Gemma Catolino, Fabio Palomba, Damian Andrew Tamburri, and Alexander Serebrenik. 2021. Understanding community smells variability: A statistical approach. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS).* IEEE, 77–86.
[9] Sébastien Cherry and Pierre N Robillard. 2004. Communication problems in global software development: Spotlight on a new field of investigation. In *International Workshop on Global Software Development, International Conference on Software Engineering, Edinburgh, Scotland.* IET, 48–52.
[10] John W Creswell and J David Creswell. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches.* Sage publications.
[11] Sadhana Deshpande, Ita Richardson, Valentine Casey, and Sarah Beecham. 2010. Culture in Global Software Development - A Weakness or Strength?. In *2010 5th IEEE International Conference on Global Software Engineering*. 67–76. https://doi.org/10.1109/ICGSE.2010.16
[12] Vahid Garousi, Michael Felderer, and Mika V Mäntylä. 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and software technology* 106 (2019), 101–121.
[13] Edward Twitchell Hall. 1989. *Beyond culture.* Anchor.
[14] Charles Hampden-Turner, Fons Trompenaars, and Charles Hampden-Turner. 2020. *Riding the waves of culture: Understanding diversity in global business.* Hachette UK.
[15] James D Herbsleb and Deependra Moitra. 2001. Global software development. *IEEE software* 18, 2 (2001), 16–20.
[16] Rashina Hoda. 2021. Socio-technical grounded theory for software engineering. *IEEE Transactions on Software Engineering* 48, 10 (2021), 3808–3832.
[17] Geert Hofstede. 1984. *Culture's consequences: International differences in work-related values.* Vol. 5. sage.
[18] Geert Hofstede. 2011. Dimensionalizing cultures: The Hofstede model in context. *Online readings in psychology and culture* 2, 1 (2011), 2307–0919.
[19] Robert House, Mansour Javidan, Paul Hanges, and Peter Dorfman. 2002. Understanding cultures and implicit leadership theories across the globe: an introduction to project GLOBE. *Journal of world business* 37, 1 (2002), 3–10.
[20] Project Management Institute. 2021. *A Guide to the Project Management Body of Knowledge* (7 ed.). 250 pages.
[21] Imran Javed, Uzair Iqbal Janjua, Shafi'i Muhammad Abdulhamid, Tahir Mustafa Madni, and Adnan Akhunzada. 2023. The Impact of Mitigation Strategies for Socio-Cultural Distance Issues in GSD: An Empirical Study. *IEEE Access* 11 (2023), 99499–99518. https://doi.org/10.1109/ACCESS.2023.3300836
[22] Mansour Javidan and Robert J House. 2001. Cultural acumen for the global manager: Lessons from project GLOBE. *Organizational dynamics* (2001).
[23] Stefano Lambiase, Gemma Catolino, Bice Della Piana, Filomena Ferrucci, and Fabio Palomba. 2024. Dealing With Cultural Dispersion: a Novel Theoretical Framework for Software Engineering Research and Practice. In *Proceedings of the 2024 ACM/IEEE 46th International Conference on Software Engineering: Software Engineering in Society.*
[24] Stefano Lambiase, Gemma Catolino, Fabiano Pecorelli, Damian A. Tamburri, Fabio Palomba, Willem-Jan Van Den Heuvel, and Filomena Ferrucci. 2022. "There and Back Again?" On the Influence of Software Community Dispersion Over Productivity. In *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA).* 177–184. https://doi.org/10.1109/SEAA56994.2022.00035
[25] Stefano Lambiase, Gemma Catolino, Fabiano Pecorelli, Damian A Tamburri, Fabio Palomba, Willem-Jan van den Heuvel, and Filomena Ferrucci. 2024. An Empirical Investigation Into the Influence of Software Communities' Cultural and Geographical Dispersion on Productivity. *Journal of Systems and Software* 208 (2024), 111878.
[26] Stefano Lambiase, Gemma Catolino, Damian A Tamburri, Alexander Serebrenik, Fabio Palomba, and Filomena Ferrucci. 2022. Good fences make good neighbours? on the impact of cultural and geographical dispersion on community smells. In *Proceedings of the 2022 ACM/IEEE 44th International Conference on Software Engineering: Software Engineering in Society*. 67–78.
[27] Marcelo Marinho, Alexandre Luna, and Sarah Beecham. 2018. Global software development: practices for cultural differences. In *International Conference on Product-Focused Software Process Improvement.* Springer, 299–317.
[28] Antonio Martini and Jan Bosch. 2017. Revealing social debt with the CAFFEA framework: An antidote to architectural debt. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW).* IEEE, 179–181.
[29] Fabio Palomba, Damian Andrew Tamburri, Francesca Arcelli Fontana, Rocco Oliveto, Andy Zaidman, and Alexander Serebrenik. 2021. Beyond Technical Aspects: How Do Community Smells Influence the Intensity of Code Smells? *IEEE Transactions on Software Engineering* 47, 1 (2021), 108–129. https://doi.org/10.1109/TSE.2018.2883603
[30] Fabio Palomba and Damian Andrew Tamburri. 2021. Predicting the emergence of community smells using socio-technical metrics: a machine-learning approach. *Journal of Systems and Software* 171 (2021), 110847.
[31] Paul Ralph, Mike Chiasson, and Helen Kelley. 2016. Social Theory for Software Engineering Research. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering* (Limerick, Ireland) *(EASE '16).* Association for Computing Machinery, New York, NY, USA, Article 44, 11 pages. https://doi.org/10.1145/2915970.2915998
[32] Ita Richardson, Valentine Casey, John Burton, and Fergal McCaffery. 2010. *Global Software Engineering: A Software Process Approach.* Springer Berlin Heidelberg, Berlin, Heidelberg, 35–56. https://doi.org/10.1007/978-3-642-10294-3_2
[33] Christian Ringle, Dirceu Da Silva, and Diógenes Bido. 2015. Structural equation modeling with the SmartPLS. *Bido, D., da Silva, D., & Ringle, C.(2014). Structural Equation Modeling with the Smartpls. Brazilian Journal Of Marketing* 13, 2 (2015).
[34] Anselm Strauss and Juliet M Corbin. 1997. *Grounded theory in practice.* Sage.
[35] Damian Andrew Tamburri. 2019. Software Architecture Social Debt: Managing the Incommunicability Factor. *IEEE Transactions on Computational Social Systems* 6, 1 (2019), 20–37. https://doi.org/10.1109/TCSS.2018.2886433
[36] Damian Andrew Tamburri, Rick Kazman, and Hamed Fahimi. 2016. The architect's role in community shepherding. *IEEE Software* 33, 6 (2016), 70–79.
[37] Damian Andrew Tamburri, Philippe Kruchten, Patricia Lago, and Hans van Vliet. 2013. What is social debt in software engineering?. In *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE).* IEEE, 93–96.
[38] Damian Andrew Tamburri, Philippe Kruchten, Patricia Lago, and Hans van Vliet. 2015. Social Debt in Software Engineering: Insights from Industry. *Journal of Internet Services and Applications* (2015). https://doi.org/10.1186/s13174-015-0024-6
[39] Damian Andrew Tamburri, Fabio Palomba, and Rick Kazman. 2019. Exploring Community Smells in Open-Source: An Automated Approach. *IEEE Transactions on Software Engineering* 47, 3 (2019), 630–652. https://doi.org/10.1109/TSE.2019.2901490
[40] Damian Andrew Tamburri, Fabio Palomba, Alexander Serebrenik, and Andy Zaidman. 2019. Discovering community patterns in open-source: a systematic approach and its evaluation. *Empirical Software Engineering* 24, 3 (2019), 1369–1417.
[41] Gianmario Voria, Viviana Pentangelo, Antonio Della Porta, Stefano Lambiase, Gemma Catolino, Fabio Palomba, and Filomena Ferrucci. [n. d.]. Community Smell Detection and Refactoring in SLACK: The CADOCS Project. ([n. d.]).
[42] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering.* Springer Science & Business Media.
[43] Affan Yasin, Rubia Fatima, Javed Ali Khan, Lin Liu, Raian Ali, and Jianmin Wang. [n. d.]. Counteracting sociocultural barriers in global software engineering using group activities. *Journal of Software: Evolution and Process* n/a, n/a ([n. d.]), e2587. https://doi.org/10.1002/smr.2587 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.2587
[44] Elijah Zolduoarrati, Sherlock A Licorish, and Nigel Stanger. 2022. Impact of individualism and collectivism cultural profiles on the behaviour of software developers: A study of stack overflow. *Journal of Systems and Software* (2022), 111427.