Investigating Cultural Dispersion: on the Role of Cultural Differences in Software Development Teams

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

ABSTRACT

Software development, inherently a social activity, involves individuals across diverse geographical and cultural settings. Despite this nature, the existing Global Software Engineering research body encounters limitations, making the achieved results challenging to use by practitioners. This Ph.D. research project seeks to overcome these constraints by crafting a theoretical framework. The framework systematically captures cultural differences, exploring their impact on various aspects of software development and delving into practitioners' strategies for managing these influences. Additionally, the project aims to significantly contribute to the professional software development landscape by transferring research findings to practitioners through practical tools. This framework serves as an immediate application for professionals, fostering project success through heightened cultural awareness and adaptability, thereby enhancing developer well-being 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 Engineering; Cultural Dispersion; Socio-Technical Aspects

ACM Reference Format:
Stefano Lambiase. 2024. Investigating Cultural Dispersion: on the Role of Cultural Differences in Software Development Teams. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3639478.3639799

1 CONTEXT AND LIMITATIONS

Software development is de facto a collaborative and human-centered activity [4, 25]; researchers in the Software Engineering community have been exploring how human and social factors—e.g., emotions and cultural background—can profoundly influence each aspect of software development. Their investigations reveal the crucial nature of managing these factors, as they wield substantial influence over the ultimate success of a software project [6, 31].

Among the various societal challenges that may potentially impact software development, *culture* is emerging as a critical element that demands attention among software community members throughout the development lifecycle. *Culture* is defined as shared motives, values, beliefs, identities, and interpretations or meanings of significant events that result from common experiences of members of collectives and are transmitted across generations [18]. Our emphasis on culture stems from a pivotal insight presented in the seminal work by Damian and Moitra [9] and by Šmite et al. [27]: culture exerts a more substantial influence on software engineering practices than initially anticipated. This underscores the primary reason why researchers in the domain of *Global Software Engineering* (GSE)—which pertains to the branch of software engineering aiming to establish a set of practices for managing globally distributed software teams—have extensively delved into the impact of cultural aspects on activities related to distributed software development and maintenance [7, 22, 29].

The key insight derived from these research papers is 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:

Contrasting Results. Literature on culture in software development presents a contrast in terms of results. On the one hand, the presence of individuals from different cultures within the same team may lead to positive long-term effects [14, 24], such as empowering decision-making processes by broadening the spectrum of opinions. On the other hand, these differences may converge into collaboration and communication problems [5, 7, 23], potentially increasing the likelihood of interpersonal conflicts. Consequently, *the role of cultural dispersion in global software engineering remains unclear and warrants further analysis.*

Lack of Knowledge on Socio-Technical Aspects. Most previous studies have explored the relationship between culture and software development by concentrating on specific processes and product metrics. For example, some studies delved into understanding how culture influences the code review process [3] or code quality [1]. However, there is still a dearth of knowledge concerning how culture may influence the behavioral patterns among developers [34], particularly in terms of communication and collaboration relationships within a development community. Consequently, *our understanding of the impact of culture on socio-technical dimensions remains limited.*

ICSE-Companion '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0502-1/24/04.
https://doi.org/10.1145/3639478.3639799

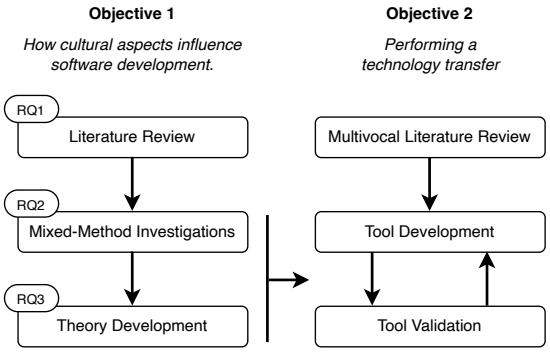

Figure 1: Research Activity Schedule.

Lack in Using Cultural Frameworks. It is noteworthy that only a limited number of studies have utilized cultural frameworks [12, 13, 16, 17], which provide (1) a set of cultural behavior of individuals associated to culture and (2) numerical values to characterize such behaviors. Moreover, previous work has predominantly treated culture as an "abstract" concept without quantifying it [22]. As a result of this approach, most of the findings reported in the literature cannot be evaluated against a reference framework, *therefore hindering a more objective understanding of how culture impacts software engineering practices.*

Lack of Theoretical Framework. Currently, no comprehensive theory adequately elucidates the impact of cultural differences on software development and effective strategies for managing these differences. This absence carries two significant implications: firstly, *it contributes to fragmented research, frequently yielding new contributions without substantial progress in the field; secondly, it constrains the practical applicability of research findings in the software development industry, as there is no established framework readily graspable and implementable by practitioners.*

Addressing the limitations above is pivotal to better understanding how cultural differences impact software development and the teams involved. Such knowledge could be bridled to empower managers to make informed decisions, consequentially increasing the success rate in software development projects and enhancing developers' well-being.

2 RESEARCH OBJECTIVE

In response to the limitations above and intending to enhance the body of knowledge in Global Software Engineering, this Ph.D. research project seeks to formulate a theoretical framework. This framework is intended to encompass the influence of cultural differences (represented in a systematic form) on various aspects of software development and explore strategies employed by practitioners to managing such impacts. Given software development's collaborative and social nature [4, 25], 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. Moreover, since software development is a socio-technical phenomenon [15, 24], a particular focus is invested in studying how socio-technical aspects are influenced by culture.

In addition to the aforementioned goal, this work aims to contribute substantially to the practitioners' software development landscape. To achieve this, we aim to turn research findings into actionable results by creating tools that embed the new knowledge.

> **Research Objective**—*Studying, understanding, and reporting on the role of cultural differences in the software development lifecycle, particularly focusing on socio-technical aspects and phenomena. Then, making such knowledge actionable by practitioners and researchers through the development of a theoretical framework and tools (e.g., recommendation systems) based on it.*

3 RESEARCH PROCESS AND EVALUATION

Figure 1 reports the research method defined to answer the research questions and address the research objectives.

3.1 First Objective—Main Research Activity

Concerning the first objective, three research questions have been formulated to guide the study [33].

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

The initial question aims to present the current state of the art on the primary topic under analysis and to pinpoint potential gaps that could contribute to the primary findings of the dissertation. Furthermore, scrutinizing the state of the art is not only aimed at providing new insights but also at laying the groundwork for original outcomes.

To address the first research question, a thorough literature review of reputable sources will be conducted, seeking secondary studies (e.g., systematic literature reviews and mapping studies) on the subject. If secondary studies are not identified, the initial step of the research involves conducting a systematic literature review and a mapping study on the role of culture in software engineering.

> **RQ$_2$**—*How do cultural differences in software development teams influence the development lifecycle?*

The second research question serves as the nucleus of the Ph.D. project, encapsulating its primary objective. It is anticipated that several more specific research queries will emanate from it and from the responses to the first question, each evolving into separate endeavors. The amalgamation of these efforts is expected to culminate in a comprehensive answer to the original question.

Concerning the second research question, the plan involves dividing the research into two phases. In the first phase, *mixed-method research approaches* [8] will be employed to explore culture in contexts identified as lacking contributions while addressing the first research question. In this research type, qualitative methods (e.g., qualitative analysis, grounded theory, and interviews) and quantitative investigations (e.g., statistically supported empirical studies and data mining) are conducted on the same dataset or in the same context to address similar research questions. The ultimate goal is to achieve *theoretical saturation*, signifying the point in category

development where no new properties, dimensions, or relationships emerge during analysis. Therefore, the theory is considered saturated if both qualitative and quantitative data converge on the same conclusions. Conversely, any disparities between the two studies would lead to the need for ulterior investigations on the matter.

> **RQ$_3$**—*Which theoretical frameworks could be used to represent and make the findings on the impact of cultural differences on software development actionable?*

In the second phase, nearing the end of the Ph.D. period, the acquired knowledge will be utilized to conduct studies aimed at developing comprehensive theoretical frameworks, representing the primary contribution of the dissertation. Both qualitative and quantitative approaches will be employed once again. The *Grounded Theory* [15, 28] approach will guide the development of theories originating from qualitative data. Specifically, a socio-technical grounded theory for data analysis [15] will be employed as it aligns seamlessly with the context under analysis. On the quantitative side, *structural equation modeling* [26] will be utilized to develop a theory supported by quantitative data. Ultimately, the two types of theories will be jointly evaluated to formulate a unified and comprehensive theory supported by both quantitative and qualitative data.

Regarding the central theme of our discussions, it is vital to acknowledge the complexities inherent in addressing culture during our investigations and the potential for misunderstandings. Deliberate steps have been taken to ensure clarity and alignment and mitigate the risk of inaccurate results. In particular, we have opted to represent culture using cultural frameworks [16, 17] rather than relying on the definition of culture, which often presents interpretation issues. These frameworks depict culture through a set of "dimensions," with each dimension (1) emphasizing differences in people's behaviors and (2) utilizing numerical values to gauge the extent of these differences.[1] Building upon 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 discussing behaviors empirically linked to them, we ensure a precise analysis, eliminating the interpretative nature associated with the definition of culture [16, 17]. This method allows us to delve into cultural dimensions through tangible behaviors established by research, ensuring clarity and minimizing subjective interpretation.

The assessment of contributions will vary depending on the context of each study. Data and statistical tools will be used to evaluate contributions during quantitative investigations. On the other side, human participants involved in the study will play a crucial role in assessing the findings. Furthermore, employing a mixed-method approach will enable the validation of results by comparing findings from both applied methods [8].

[1]For instance, Hofstede's *Uncertainty Avoidance* dimension assesses a society's comfort with uncertainty [16]. High scores indicate a preference for rules and stability, avoiding ambiguity, while low scores suggest openness to change, risk-taking, and adaptability to uncertainty.

In light of our frequent involvement with human participants, we will prioritize ethical considerations. An ethical review board will be consulted, particularly for studies requiring ethical oversight. This ensures our research adheres to rigorous ethical standards, safeguarding participants' well-being, rights, and privacy.

3.2 Second Objective—Technology Transfer

For the second objective, given that the primary aim is to develop tools for turning research findings into actionable results, a comprehensive literature review with multiple perspectives will be undertaken. The aim is to (1) pinpoint the optimal instrument for executing technology transfer and (2) comprehend the current state of the art regarding best practices and challenges in its development and adoption. A *multivocal literature review* (MLR) [10] will be conducted rather than a systematic one; MLR is ideally indicated to study aspects related to practitioners and provide results beneficial for them [10]. After establishing foundational knowledge, a series of tools will be developed and iteratively refined with the goal of offering practitioners and researchers a user-friendly and accessible means to apply insights from leading research. These tools will be presented as tool demonstration papers and disseminated through various social channels.

In terms of evaluation, a dual approach will be adopted. First, as the tools are designed for practical use, a usability evaluation involving human participants will be conducted [11]. Additionally, each tool will undergo evaluation regarding its efficacy in enabling the correct application of research findings by individuals. This evaluation will be tailored to the specific case and will encompass both human and statistical assessments.

4 EXPECTED CONTRIBUTION

The primary contributions of the research align with its objectives, encompassing two key aspects.

Theory of Dealing with Cultural Dispersion. Firstly, a theoretical framework will be developed, delving into the impact of individual culture on software development's socio-technical aspects and how professionals navigate this dynamic. This framework, grounded in both qualitative and quantitative studies, will address limitations identified in the current state of the art and will be designed to benefit both practitioners and researchers.

Tools for Performing Technology Transfer. The second contribution involves a set of tools aimed at making research findings easily usable and widely accessible to all.

Together, these contributions aim to advance understanding in the field, providing practical insights for professionals and fostering widespread dissemination of research outcomes. Moreover, the identified limitations in the state of the art will be addressed.

In order to address the limitation on socio-technical aspects, since software development is a socio-technical phenomenon [15], the primary goal is to investigate the impact of culture on socio-technical aspects (e.g., technical and social debt). For instance, a key focus of our investigation revolves around examining the influence of cultural dispersion on *community smells* [30], i.e., socio-technical anti-patterns within the collaborative structure of a development community that are precursors of social debt [30].

Moreover, as we deal with cultural differences, we prefer using cultural frameworks [16, 17] instead of getting stuck in the tricky

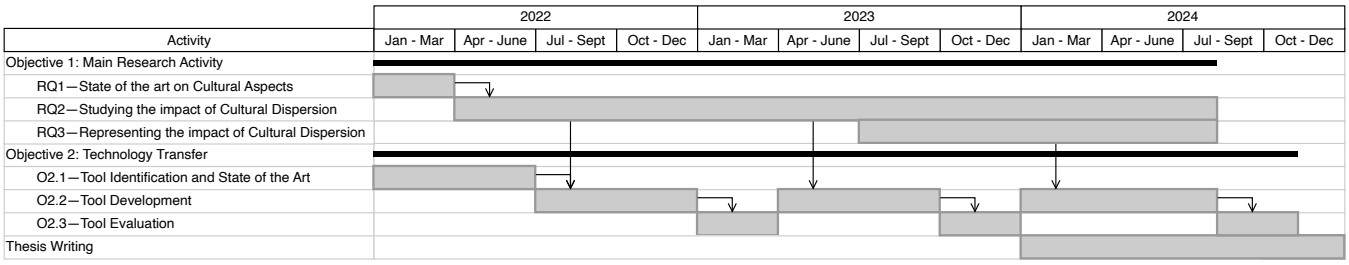

Figure 2: Research Activity Schedule.

task of defining culture, which can lead to confusion. We put these frameworks into action in two ways: by talking about culture based on behaviors (the qualitative side) and by using the numbers tied to these cultural dimensions (the quantitative side). This helps us avoid being too vague in our findings.

Lastly, by conducting concurrent qualitative and quantitative experiments and comparing their outcomes, the aim is to formulate theories that comprehensively and robustly depict the current state of art and practice [8, 15, 26]. This approach seeks to (1) unveil the role of culture in software development and (2) provide theoretical frameworks usable by both professionals and researchers.

By mitigating the identified limitations, it is possible to formulate a practical, theoretical framework that delineates the influence of cultural differences on software development. This tool seeks to elevate project success rates by promoting heightened cultural awareness and adaptability, ultimately fostering improved well-being among developers in diverse environments.

5 PRELIMINARY RESULTS AND SCHEDULING

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

Results Achieved for the Main Research. Addressing the first objective, the literature analysis identified limitations, detailed in the paper's introduction, guiding subsequent steps and answering the first research question.

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 [19–21]:

- The investigation into the impact of culture on the social aspects of software development informed the first study. We hypothesized that cultural dispersion could influence collaboration, leading to the emergence of community smells. The study showed that cultural dispersion affects the emergence of all community smells, with nuanced results—indicating not solely negative impacts. A notable finding was the correlation between the presence of individualistic and collectivist individuals and the emergence of Lone Wolf effects. The work [21] was published at the *International Conference on Sostware Enginnering-Software Engineering in Society* (ICSE-SEIS 2022).
- Expanding our focus on socio-technical metrics, we delved into the relationships between cultural dispersion and productivity. A mixed-method study revealed that dispersion metrics can both positively and negatively affect productivity, contingent on how managers address cultural differences. For instance, integrating individualistic and collaboration-oriented individuals might prove ineffective, underscoring the need for a nuanced approach. This work resulted in two publications [19, 20]: the first one [19] at the *Software Engineering and Advanced Applications Euromicro Conference* (SEAA 2022) and the second [20] (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.

In the same context, additional work has been submitted and is currently under review to conclude the first objective through the third research question.

Results Achieved for the Technology Transfer. Regarding the second objective, we identified conversational agents, commonly known as chatbots, as a managerial tool within the context of software development. Initially, we conducted a comprehensive literature review (currently under revision) to accumulate insights into the adoption and challenges associated with these tools in the software engineering domain.

Building upon the acquired knowledge, we developed CADOCS (**C**onversational **A**gent for the **D**etection **O**f **C**ommunity **S**mells) [32], a conversational agent published 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) enhance its usability within a well-established communication channel like Slack, and (2) augment its functionality by offering 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 seamless integration of emerging insights from cutting-edge research.

In terms of future work, as Illustrated in the Gantt chart in Figure 2, our current research emphasis lies in the validation of new tools and the formulation of theories. Substantial work has been dedicated to qualitative theory development, and the construction of theory based on quantitative data (SEM) is underway, with a planned submission in the upcoming months. Following completion, a CADOCS tool update will be crafted, integrating insights derived from the conducted studies. Ultimately, the thesis writing phase is scheduled to commence in early January 2024.

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] 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.

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

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

[6] Narciso Cerpa and June M. Verner. 2009. Why Did Your Project Fail? *Commun. ACM* 52, 12 (Dec. 2009), 130–134. https://doi.org/10.1145/1610252.1610286

[7] 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.

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

[9] Daniela Damian and Deependra Moitra. 2006. Guest editors' introduction: Global software development: How far have we come? *IEEE software* 23, 5 (2006), 17–19.

[10] 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.

[11] Alex Genov. 2005. Iterative usability testing as continuous feedback: A control systems perspective. *Journal of Usability Studies* 1, 1 (2005), 18–27.

[12] Edward Twitchell Hall. 1989. *Beyond culture*. Anchor.

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

[14] James D Herbsleb and Deependra Moitra. 2001. Global software development. *IEEE software* 18, 2 (2001), 16–20.

[15] Rashina Hoda. 2021. Socio-technical grounded theory for software engineering. *IEEE Transactions on Software Engineering* 48, 10 (2021), 3808–3832.

[16] Geert Hofstede. 2011. Dimensionalizing cultures: The Hofstede model in context. *Online readings in psychology and culture* 2, 1 (2011), 2307–0919.

[17] 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.

[18] Mansour Javidan and Robert J House. 2001. Cultural acumen for the global manager: Lessons from project GLOBE. *Organizational dynamics* (2001).

[19] 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

[20] 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.

[21] 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.

[22] 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.

[23] John Noll, Sarah Beecham, and Ita Richardson. 2011. Global software development and collaboration: barriers and solutions. *ACM inroads* 1, 3 (2011), 66–78.

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

[25] 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

[26] 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).

[27] Darja Šmite, Claes Wohlin, Tony Gorschek, and Robert Feldt. 2010. Empirical evidence in global software engineering: a systematic review. *Empirical software engineering* 15, 1 (2010), 91–118.

[28] Anselm Strauss and Juliet M Corbin. 1997. *Grounded theory in practice*. Sage.

[29] Viktoria Stray and Nils Brede Moe. 2020. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. *Journal of Systems and Software* 170 (2020), 110717.

[30] 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

[31] Damian A. Tamburri, Fabio Palomba, and Rick Kazman. 2021. Success and Failure in Software Engineering: A Followup Systematic Literature Review. *IEEE Transactions on Engineering Management* 68, 2 (2021), 599–611. https://doi.org/10.1109/TEM.2020.2976642

[32] 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.]).

[33] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[34] 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.