Socio-Technical Well-Being of Quantum Software Communities: An Overview on Community Smells

Stefano Lambiase[0000−0002−9933−6203], Manuel De Stefano[0000−0001−6038−4171], Fabio Palomba[0000−0001−9337−5116], Filomena Ferrucci[0000−0002−0975−8972], and Andrea De Lucia[0000−0002−4238−1425]

University of Salerno, Fisciano (SA), Italy
{slambiase,madestefano,fpalomba,fferrucci,adelucia}@unisa.it

Abstract. Quantum computing has gained significant attention due to its potential to solve computational problems beyond the capabilities of classical computers. With major corporations and academic institutions investing in quantum hardware and software, there has been a rise in the development of quantum-enabled systems, particularly within open-source communities. However, despite the promising nature of quantum technologies, these communities face critical socio-technical challenges, including the emergence of socio-technical anti-patterns known as community smells. These anti-patterns, prevalent in open-source environments, have the potential to negatively impact both product quality and community health by introducing technical debt and amplifying architectural and code smells. Despite the importance of these socio-technical factors, there remains a scarcity of research investigating their influence within quantum open-source communities. This work aims to address this gap by providing a first step in analyzing the socio-technical well-being of quantum communities through a cross-sectional study. By understanding the socio-technical dynamics at play, it is expected that foundational knowledge can be established to mitigate the risks associated with community smells and ensure the long-term sustainability of open-source quantum initiatives.

Keywords: Quantum Software Enginering · Socio-Technical Aspects · Community Smells · Open Source Communities.

1 Introduction

In recent years, *quantum computing*, a field of computer science based on quantum theory, has gained significant attention in both research and industry. Quantum software technologies are increasingly adopted for their potential to solve computational problems beyond the capabilities of classical computers [13, 16]. As a result, major companies like IBM and Google have invested heavily in quantum hardware, offering users access to resources for experimentation and development. This has enabled the creation of *quantum-enabled systems* that integrate quantum software into their operations, offering significant benefits

for software development. For example, the exponential increase in processing power from quantum computing [43] could lead to higher-quality software products, while *quantum machine learning (QML)* supports deeper analysis of large, complex datasets, advancing both machine learning and scientific research.

To democratize access to this transformative technology, the field of *quantum software engineering (QSE)* has emerged [19, 23–25]. Since the publication of the Talavera Manifesto, researchers have been actively designing and implementing advanced quantum software applications that leverage the computational capabilities of quantum computers [42]. In parallel, significant efforts have been made to keep quantum software development largely an open-source practice [26, 27, 31, 39]. This approach is reinforced by academic studies that explore this phenomenon from different perspectives [7, 31]. An important observation is that today, it can be observed that the majority of the community involved in quantum software development operates within open-source contexts, as evidenced by the numerous repositories and contributors across various collaboration platforms [26, 27, 31, 39].

However, the advantages of open-source activities come with significant socio-technical challenges. Extensive research has examined the socio-technical dynamics within open-source communities, resulting in the identification and classification of various socio-technical anti-patterns, which highlight problems in the organizational and collaborative structures of these communities [5, 20, 33, 35]. These issues can have serious consequences, potentially leading to critical project failures and, in the worst scenarios, the death of the community. Indeed, software development, by its nature, is a socio-technical activity [4, 14, 28], involving stakeholders from diverse backgrounds who collaborate on innovative technologies. Poor management of this diversity can lead to subtle, often unnoticed social problems, eventually contributing to what is termed *social debt*, the hidden costs of maintaining a development community with suboptimal dynamics [35]. Researchers' work on identifying and addressing these problems has led to the concept of *community smells*, socio-technical anti-patterns (e.g., excessive formality) and behavioral patterns (e.g., repeated condescension or sudden departures) that can exacerbate social debt [34, 36]. These anti-patterns should not be underestimated. Research has demonstrated that community smells, particularly in open-source environments, have the potential to significantly harm both the community and the product. For instance, they can negatively impact product quality and introduce technical debt through the amplification of both architectural [32] and code smells [20, 35].

Given that quantum software development is primarily conducted as an open-source activity, it can be argued that quantum communities are not immune to socio-technical anti-patterns like community smells. As a result, these communities may face socio-technical challenges that could impede their progress, which is particularly crucial during this pivotal phase of technological innovation that quantum computing and development are experiencing. Despite the relevance of these concerns, also reported by relevant literature [10, 31], **studies specifically investigating the socio-technical dimensions of quantum open-source**

communities are entirely absent. This lack of research is problematic for several reasons. First, without socio-technical insights, it becomes difficult to identify and address challenges that could slow the development and adoption of quantum technologies, which are heavily dependent on collaborative efficiency. Second, the absence of research in this area limits the ability of community leaders to foster inclusive and sustainable environments, potentially leading to the exclusion of diverse talents that are vital for innovation. Finally, as quantum technologies continue to grow rapidly, unresolved community smells risk accumulating into social debt, which could increase long-term project costs and hinder the scalability of open-source quantum initiatives.

To address the limitations mentioned above, we aimed to provide a first step into the investigation of the socio-technical well-being of quantum software communities by means of a statistical overview. By doing so, we aimed to depict the current situation of open-source communities developing quantum-enabled projects to put foundational knowledge and reasons for ulterior works in such a context. It is reasonable that this knowledge is essential: as highlighted earlier, the presence of socio-technical anti-patterns (like community smells) has been correlated with, and shown to influence, the emergence of other issues not only of a social nature but also strictly technological (such as architectural and code smells [20, 32, 35]). These issues have the potential to undermine a software community, significantly limiting, and in some cases completely nullifying its impact.

To achieve our goal, we carried out a *cross-sectional study* [11,38], which is a type of research designed to assess and illustrate essential traits of a population at a particular moment in time. Such studies offer a snapshot of the occurrence of a disease or condition (community smells) and the spread of various factors among a population (communities developing quantum-enabled systems).

2 Background and Related Work

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

2.1 Quantum Computing and Quantum Software Communities

Quantum computing is a field within computer science that leverages the principles of quantum theory, specifically applying quantum mechanics to perform computations [13,16]. Unlike classical computing, which relies on bits that take binary values (zero or one), quantum computing uses qubits, which can exist in a superposition of both zero and one states simultaneously. Quantum gates, which perform unitary transformations, are used to manipulate quantum information. Quantum programming languages manage both classical and quantum data using registers, and quantum programs are represented as quantum circuits, where gates are applied in a specific sequence. These circuits are executed on either

real quantum hardware or simulators, with the results measured and stored in classical registers.

Quantum Software Engineering (QSE) is an emerging research field that has been formalized through the *Talavera Manifesto* [24], which established the foundational principles for this discipline. In extending this contribution, Piattini et al. [25] focused on defining specific research domains within QSE. They identified four areas: the design of quantum-hybrid systems, testing methodologies, assessing the quality of quantum programs, and re-engineering classical-quantum information systems. A key point raised in their work is the need to bridge the knowledge gap between quantum computer scientists and traditional software engineers, as collaboration between these fields is essential for advancing QSE.

Building on this work, De Stefano et al. [9] conducted a systematic mapping study to reveal the current state of research in QSE. Their study [9] found that the primary focus of QSE research has been on software testing, highlighting a concentration of efforts in this area. However, their findings also suggest the need for a more balanced distribution of research efforts across other QSE domains. In particular, socio-technical aspects within quantum software development communities and software engineering management have received less attention, despite growing recognition of their importance. Both contributors [39] and researchers [10, 31] have called for increased research in these areas to address the unique challenges posed by the interdisciplinary nature of quantum software development.

2.2 Socio-Technical Well-Being—Community Smells

Software development and its engineering are inherently socio-technical activities. To assess the influence of social dynamics on software development, researchers—drawing on the well-established notion of Technical Debt [20, 21]—introduced the concept of *Social Debt*, which refers to the unforeseen costs associated with sub-optimal decisions in collaboration, communication, and team management [5, 35]. Additionally, in an effort to further characterize and identify the sources of Social Debt, researchers proposed the concept of *Community Smell*, defined as socio-technical anti-patterns that may negatively impact the socio-technical well-being of a software development team, potentially leading to the accumulation of Social Debt [5].

The research explored the relationship between community smells and various aspects of software development. Notably, Palomba et al. [20] examined the connection between community smells and code smells, their product-oriented counterpart, demonstrating that community smells are among the primary factors influencing the emergence of code smells. Tamburri et al. [33] conducted a large-scale study across 60 open-source ecosystems to assess (1) the diffusion of community smells and (2) their perceived impact by developers, revealing that community smells are both widespread and perceived to affect the evolution and sustainability of software communities. These findings indicate that socio-technical antipatterns can influence maintenance and evolution in two ways: by directly increasing social debt (i.e., increasing costs related to socio-technical

problems) and by affecting product-related factors, thereby increasing technical debt. Furthermore, two mining studies conducted by Catolino et al. [6] and Lambiase et al. [17] revealed correlations between the emergence of community smells and gender diversity (in the former) and cultural heterogeneity (in the latter).

Regarding detection methods, Palomba and Tamburri [37] proposed a machine learning approach for predicting community smells based on socio-technical metrics, achieving promising results with an F-measure of 78%. Additionally, Almarimi et al. [2] introduced a multi-label learning model using genetic algorithms to detect ten community smells and developed the community smells detection tool csDetector [1]. Building on Almarimi et al.'s work [1, 2], Voria et al. [40] developed CADOCS, a conversational agent capable of detecting ten community smells from a software repository and proposing refactoring strategies for some of them. Moreover, Advanced network models, such as the MOGen higher-order network model, have been developed to detect community smells by analyzing complex relationships and interaction patterns within software teams [12].

Related Work: Summary and Research Gap.

The effort to keep quantum computing tied to open source has not been matched by efforts to understand the socio-technical challenges of these communities. This gap limits contributors' ability to assess community health and restricts QSE researchers from exploring a well-established area of software engineering.

3 Cross-sectional Study—An Overview

This section provides the reader with basic knowledge of observational studies [38, 41]. For space limitations, we suggest reading the work of Saarimäki et al. [29, 30] to gain a better understanding of the method.

3.1 Cross-Sectional Studies

Observational studies, including *cross-sectional studies*, are widely used in epidemiology to examine associations between exposures and outcomes without intervention [38, 41]. The main types include *cohort*, *case-control*, and *cross-sectional* designs [38, 41].

Cross-sectional studies capture population characteristics at a single time point, providing a snapshot of prevalence and exposure distribution [41]. Data are collected simultaneously from all participants, enabling efficient assessment of existing conditions. *Prevalence*—the proportion of individuals with a given condition—is central to these studies [41]. The *Prevalence Odds Ratio (POR)* quantifies the association between an exposure and a condition by comparing the odds of exposure in affected versus unaffected individuals [38, 41].

Cross-sectional studies are time- and cost-efficient and useful for estimating prevalence and generating hypotheses [38, 41]. However, they cannot determine causality or temporal order and may be affected by biases from self-reported data.

Table 1. Community Smells investigated in our study.

Community Smell	Definition
Organizational Silo (OSE)	Siloed areas of the community that do not communicate, except through one or two of their members.
Black Cloud (BCE)	Information overload due to a lack of structured communications or cooperation governance.
Radio Silence (RS)	One interposes herself into every formal interaction across more sub-communities with little flexibility to introduce other channels.
Prima Donnas (PDE)	A team member is unwilling to respect external changes from other team members.
Sharing Villainy (SV)	Cause of a lack of information exchange, team members share essential knowledge such as outdated, wrong, and unconfirmed information.
Organizational Skirmish (OS)	A misalignment between different expertise levels of individuals involved in the project leads to dropped productivity and affects the project's timeline and cost.
Solution Defiance (SD)	The development community presents different levels of cultural and experience background, leading to the division of the community into similar subgroups with completely conflicting opinions.
Truck Factor Smell (TF)	Risk of significant knowledge loss due to the turnover of developers resulting from the fact that project information and knowledge are concentrated in a minority of the developers.
Unhealthy Interaction (UI)	Long delays in stakeholder communications cause slow, light and brief conversations and discussions.
Toxic Communication (TC)	Communications between developers are subject to toxic conversations and negative sentiments containing unpleasant, anger or even conflicting opinions towards various issues that people discuss.

3.2 Mining Software Repositories as Observational Studies

The Software Engineering research community has seen a significant increase in studies that use mining software repositories (MSR), with the rise in popularity of online code repository platforms like GitHub. This has led to the introduction

of rules of thumb, highlighting common pitfalls [15] to improve the quality of these studies and their outcomes.

However, it is important to note that these studies cannot provide causal explanations for observed phenomena, despite their usefulness and straightforward execution. To address this limitation, Saarimäki et al. [29, 30] recommended the adoption of observational studies, particularly cohort studies, which offer the highest level of scientific evidence.

In the context of QSE, a relatively emerging discipline with limited guidelines and tools for investigating socio-technical issues, a cross-sectional study is justified for several reasons. Firstly, guidelines for MSR studies are somewhat limited [29], and pitfalls abound [15]. Secondly, in the field of QSE, socio-technical aspects have not been extensively explored [9]. A cross-sectional study hence represents a pragmatic and cost-effective initial step to spark the investigation of socio-technical aspects in QSE, which in this context are represented by community smells and the relationships among them.

Research on socio-technical aspects of QSE is a new area that has not been explored yet [9]. Cross-sectional studies can provide valuable insights into this field's current conditions and associations. These studies are especially helpful in generating hypotheses and exploring potential connections. Despite the limitations, cross-sectional studies provide a practical starting point for further research.

4 Research Design

The *goal* of this research was to examine the socio-technical well-being of open-source software communities developing quantum software enabled-systems. The *purpose* was to uncover foundational knowledge able to (1) inform open-source contributors' future choices and (2) shed light on future research agenda on socio-technical aspects in the field of QSE.

In order to reach our objective, we operationalized the socio-technical well-being of open-source communities using community smells. At first, we wanted to understand the current situation of such communities, aiming to depict the current diffusion of community smells. Identifying how widespread community smells are in quantum computing projects helps to highlight socio-technical challenges that can impact the productivity and health of these open-source communities. Thus, we formulated the following research question:

> ⍰ **RQ$_1$**: *What is the prevalence of community smells in quantum projects?*

After assessing the diffusion of smells, in order to better characterize the socio-technical well-being of open-source quantum software communities and the phenomenon of community smells inside them, we investigated the correlation between the different smells in the same community. Exploring relationships between different community smells can reveal deeper socio-technical issues, helping us understand how these problems interact and impact community dynamics. Thus, we formulated the following second research question:

> ⍰ **RQ$_2$**: *Is there any relationship between different community smells in the context of quantum projects?*

To answer the research questions, we conducted a cross-sectional study. First, we selected a set of quantum-enabled software projects on GitHub and extracted community smells from them. Then, to answer RQ_1, we computed the *prevalence* of the community smells, while, to answer RQ_2, we computed the *Prevalence Odds Ratio* to assess the correlations between community smells. Further details about our research process are in the following sections and in our online appendix [18].

Table 2. Metrics for the 17 analyzed repositories.

Repository	Contributors	Commits	Stars	Start Date
qrand	3	287	22	2020-10-14
bloch_sphere	3	27	76	2020-06-01
tweedledum	6	263	86	2018-07-13
xacc	21	2546	138	2017-09-19
tequila	35	1313	305	2020-04-28
qsearch	3	895	29	2019-05-29
quantpy	3	20	13	2017-09-28
QTensor	6	453	40	2018-07-23
quantum_decomp	2	77	21	2019-05-06
node-red-contrib-quantum	7	148	13	2021-06-16
nanite	11	667	14	2012-01-24
scikit-quant	3	220	34	2019-01-10
dc-qiskit-qml	2	85	10	2019-01-13
quantum-robot	3	199	4	2020-06-22
OpenFermion-FQE	10	404	42	2020-04-01
shor	4	80	8	2020-02-19
OpenFermion-Cirq	15	254	267	2018-03-20

4.1 Population and Data Collection

To answer our research questions, we investigated the contributors of communities that actively participate in the development of open-source quantum-related software. Thus, we chose to focus on a representative subset of the target population due to logistical limitations in collecting information across all quantum open-source projects on GitHub.

We took advantage of a previous dataset of 115 repositories published in a paper by De Stefano et al. [8]. The number of contributors per repository ranges

from just 1 to a maximum of 10. This indicates that most repositories have a small core team, while a few involve slightly larger groups. The number of commits varies widely, with a median value of 65, indicating significant differences in development activities across repositories. The number of stars, used as an indicator of popularity, also shows considerable variability, with a median of 10 stars per repository, suggesting a generally modest level of community recognition.

After selecting the communities for analysis, community smells were computed for each one project. The augmented version of csDetector [1],[1] available in the main repository of CADOCS [40], was used for this task. It is important to note that the tool can detect ten types of community smells, reported in Table 1. Specifically, the tool was applied with its default parameters to each repository in the selected dataset of quantum open-source repositories. To work properly, csDetector requires the URL of the repository and that certain criteria are met.[2]

Ultimately, 17 repositories (described in Table 2) out of the original 115 were successfully analyzed.[3] These repositories have varying numbers of contributors, ranging from 2 to 35, reflecting different scales of community involvement. The number of commits ranges from 20 to 2546, indicating diverse levels of development activity. The repositories also have different levels of community recognition, as shown by their stars, which range from 4 to 305. The opening and last commit dates provide insights into the lifecycle of each repository, showcasing both long-standing and more recent projects.

4.2 Data Analysis

To determine the prevalence of each community smell (thus, to answer the first research question), we computed the prevalence denoted as $P(X)$ [41], which represents the ratio of repositories where a specific smell was identified to the total number of repositories analyzed: $P(X) = \frac{\text{Repositories with Smell X}}{\text{Total Repositories}}$

To assess correlations among community smells (thus, to answer RQ_2), we employed the POR as a statistical tool [41]. It evaluates how the presence of one condition (X_1) correlates with the presence or absence of another condition (X_2)—community smells, in our context. The formula for POR is $POR = \frac{AD}{BC}$ [41], where: A is the number of cases where both conditions (X_1 and X_2) are present, B is the number of cases where condition X_1 is present, but condition X_2 is absent, C is the number of cases where condition X_1 is absent, but condition X_2 is present, and D is the number of cases where both conditions (X_1 and X_2) are absent.

[1] csDetector augmented: `https://github.com/gianwario/csDetector`

[2] csDetector criteria include (1) the presence of commits, (2) the existence of multiple authors, (3) the availability of a main branch, (4) at least one pull request (or more), (5) ensuring that the messages associated with these pull requests are not empty, and (6) at least one issues (or more).

[3] It is important to note that various state-of-the-art tools were tested to detect community smells [22, 37], but csDetector was the only one capable of analyzing a subset of the original dataset.

To further interpret the POR results, values significantly above 1 suggest a strong positive association between the community smells. For example, a POR of 2 would indicate that community smell X_1 is twice as likely to occur in communities where community smell X_2 is present, compared to those where it is absent. On the other hand, a POR less than 1 indicates a negative correlation, meaning the presence of smell X_2 decreases the likelihood of smell X_1. For instance, a POR of 0.5 would imply that community smell X_1 is half as likely to occur in the presence of smell X_2, suggesting that smell B may play a mitigating role.

4.3 Threat to Validity

The study acknowledges the presence of threats to validity that could impact the integrity of the findings.

Regarding *construct validity*, the study's scope is limited to a subset of known community smells due to tool limitations, which could potentially overlook certain socio-technical challenges within quantum developer communities. However, despite this limitation, the study highlights the importance of considering community smells in software development. It is worth noting that every component in a repository may be affected by community smells, and while the assumption made in the study may not be entirely accurate, it serves as a useful starting point. Furthermore, the technical limitations of the employed tools should be taken into account when interpreting the study's results. Overall, this study provides valuable insights into the impact of community smells on software development while also highlighting the need for further research in this area.

Concerning *conclusion validity*, the cross-sectional nature of the study design poses a threat to drawing causal conclusions between exposure factors (community smells) and outcomes. The absence of longitudinal data precludes establishing temporal relationships or causal links.

Regarding *internal validity*, a potential selection bias arises from the reliance on repositories analyzed in a previous study, limiting the generalizability of results to the entire quantum open-source landscape. Additionally, the application of the CSDETECTOR tool introduces the possibility of measurement bias, as not all community smells may be detected.

Concerning *external validity*, the study's findings cannot be universally generalized to all quantum open-source repositories, as the sample was selected based on a previous study and analyzed using a specific tool. Therefore, the broader quantum developer community may not be fully represented.

5 Results

In this section, we present the results of our analysis that we conducted and described in Section 4.

Figure 1 depicts the prevalence of community smells that we found in the sample. What is most evident is that more than half of the considered smells—i.e.,

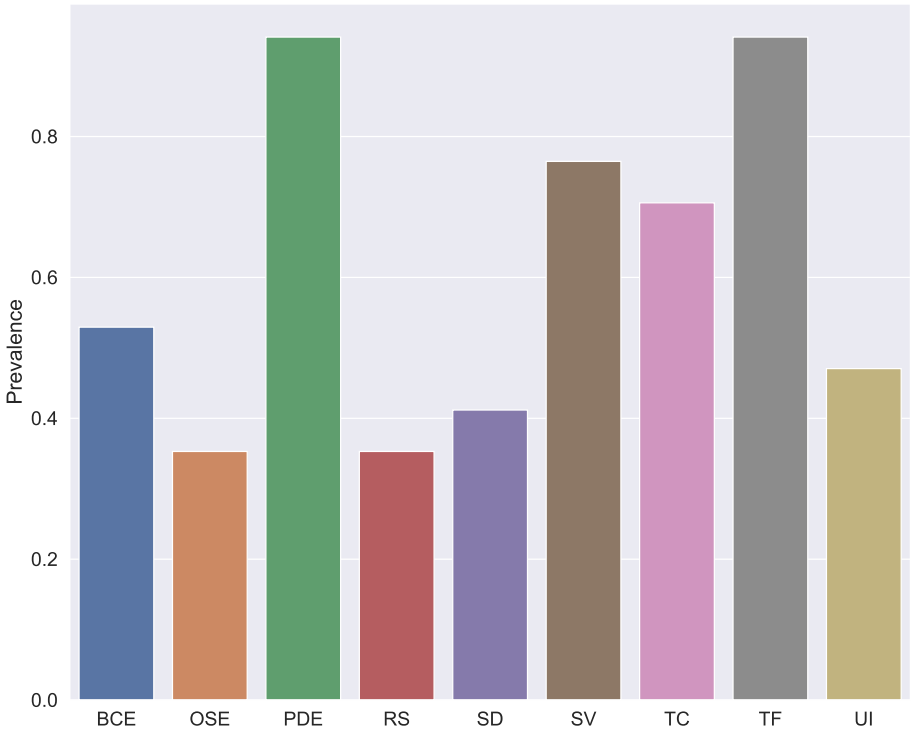

Fig. 1. Prevalence of community smells affecting the sampled repositories.

Black Cloud Effect (BCE), Power Distance Effect (PDE), Sharing Villainy (SV), Toxic Communication (TC), and Truck Factor (TF)—have a prevalence of more than 50%. This implies that these particular smells are pervasive within the analyzed repositories, with more than half of the repositories exhibiting them. The most extreme values are shown by PDE and TF, which occur in 94% of the cases. On the flipped side, Radio Silence (RS) and Organizational Silo Effect (OSE) are the least occurring smells, both exhibiting a prevalence of 35%, while Solution Defiance (SD) exhibits a prevalence of 40%. What is also interesting is that Organizational Skirmish (OS) is the only smell that never occurs in any of the considered repositories.

These findings provide a direct answer to RQ_1, showing that several community smells—most notably PDE and TF—are highly prevalent in quantum projects, with over half of the repositories affected by at least five different smells.

In Figure 2, we can see the matrix of the POR that affects the considered repositories. What can be immediately seen is that there is no smell that has only positive correlations with the other smells. On the contrary, there are some smells that have only negative correlations. Starting from the positive correlations, we discovered some interesting patterns from our analysis. For instance, if RS is

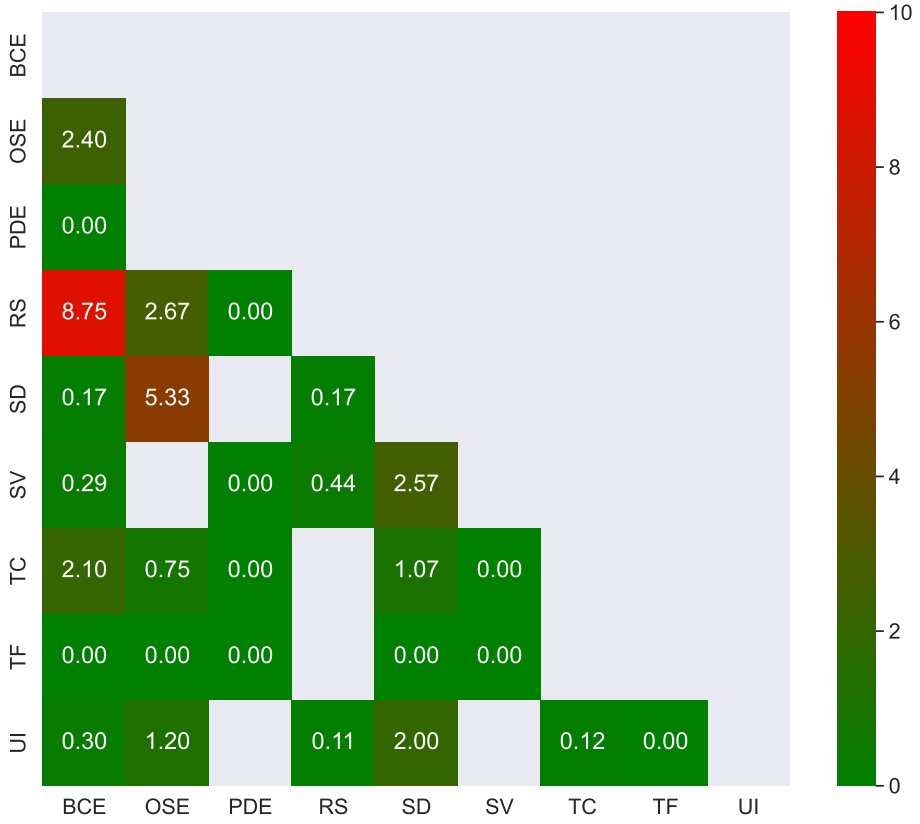

Fig. 2. Prevalence Odds Ratio among the community smells affecting the sampled repositories.

present in a repository, it is highly likely to encounter BCE, as they have a strong association (POR 8.750). Similarly, if a repository has SD, it is more likely to feature OSE, as they have an association (POR 5.333). We also found moderate positive correlations between several other community smells: RS and OSE have a POR of 2.667, SV and SD have a POR of 2.571, OSE and BCE have a POR of 2.400, and TC and BCE have a POR of 2.100. The presence of UI moderately associates with the presence of SD (POR 2.000) but only weakly associates with OSE (POR 1.200). Lastly, we observed no correlation between TC and SD (POR 1.07).

The observations where the POR is below 1 clearly indicate a significantly lower likelihood of co-occurrence. For example, we noted a POR of 0.75 which indicates a negative correlation between TC and OSE. This means that repositories containing TC are less likely to have OSE. Similarly, when compared to RS, SV exhibited a POR of 0.444, which indicates a reduced likelihood of finding RS in repositories featuring SV. We also discovered additional negative correla-

tions. Notably, SD and BCE exhibit a POR of 0.171, signifying a strong negative correlation. This indicates that when SD is present, repositories are significantly less likely to exhibit BCE. Similar negative correlations were identified between SD and RS, as well as UI and BCE, TC, and RS. Interestingly, a few pairs exhibited a POR of 0.0, suggesting a complete absence of correlation between these community smells. These pairs included various combinations, such as PDE and BCE, TF and PDE, TF and SV, TF and SD, TC and SV, TF and OSE, TF and BCE, RS and PDE, TC and PDE, SV and PDE, and UI and TF.

In summary, these findings address RQ_2 by confirming that community smells in quantum projects exhibit both strong positive and negative relationships, with some pairs frequently co-occurring and others rarely appearing together.

6 Discussions and Implications

This section reports some discussion points and implications that better contextualize our findings and could open up future work in the context of quantum software communities.

6.1 RQ_1: On the Prevalence of Community Smells

The results demonstrate that community smells, particularly Black Cloud (BCE), Prima Donnas (PDE), Sharing Villainy (SV), Toxic Communication (TC), and Truck Factor (TF), are highly prevalent within quantum software projects. With over half of the repositories exhibiting these smells, the data suggest that these issues are embedded in the fabric of quantum open-source communities. Notably, the extreme prevalence of PDE and TF, at 94%, indicates structural problems in collaboration and knowledge retention.

Compared to classical open-source communities, as explored by Tamburri et al. [33] and reported by Caballero-Espinosa et al. [5], quantum repositories exhibit a distinct pattern of smell manifestation. While classical settings often report Bottleneck, Lone Wolf, and Organizational Silo, such smells rarely reach similar extremity. These differences likely stem from the relative immaturity of quantum ecosystems, which are shaped by niche expertise, smaller contributor bases, and tightly coupled development workflows. A cross-domain comparison with ML-enabled systems further supports this interpretation. Building on the work by Annunziata et al. [3], which shares the same detection strategy as ours, both PDE and TF again emerged as dominant, confirming that knowledge concentration and authority imbalance are recurring issues in high-tech domains. However, quantum repositories exhibited higher prevalence of TC, possibly reflecting the greater communicative friction introduced by the abstract and rapidly evolving nature of quantum computing. These patterns underscore the necessity of domain-aware socio-technical strategies when managing emerging software ecosystems.

Our findings have significant implications both for researchers and contributors to these projects:

- For *researchers*, the results highlight the need for targeted studies on the role of community smells—particularly PDE and TF—in shaping the sustainability of quantum software projects. These smells were not only highly prevalent but more extreme than in classical or ML-enabled systems, suggesting domain-specific coordination challenges. Future work should investigate whether such smells hinder innovation or community growth and explore interventions like improved governance or communication structures to mitigate them. This would extend socio-technical theory into the context of emerging, expertise-driven ecosystems.
- For *open-source contributors*, the prevalence of PDE and TF indicates risks related to centralization and knowledge retention. While smaller teams and uniform expertise—reflected in the absence of OS—may ease collaboration, they also require proactive practices to prevent silos and ensure continuity. Lightweight mentoring, onboarding strategies, and documentation can help reduce these risks and support healthier project evolution.

6.2 RQ_2: On the Relationship Between Community Smells

The results reveal a complex network of both positive and negative correlations between community smells in quantum software projects, providing valuable insights for both researchers and open-source contributors. The significant positive correlations, such as those between Radio Silence (RS) and Black Cloud (BCE) (POR 8.750), as well as Solution Defiance (SD) and Organizational Skirmish (OSE) (POR 5.333), suggest that these particular smells often co-occur. This may indicate systemic issues in communication and collaboration, which reinforce each other. For instance, the strong link between RS and BCE highlights the potential for communication breakdowns leading to information overload, while the connection between SD and OSE points to the misalignment of expertise exacerbating silos within the community.

On the other hand, the negative correlations, such as that between TC and OSE (POR 0.75), suggest that certain smells are less likely to appear together. This could imply that when one issue is present, it mitigates or prevents the formation of others, possibly due to counterbalancing effects in community dynamics. The negative correlation between SD and BCE (POR 0.171) supports this idea, suggesting that repositories struggling with conflicting opinions and expertise may paradoxically avoid issues related to unstructured communication.

Unlike classical open-source systems, for which—to the best of our knowledge—no existing studies analyze correlations between community smells, ML-enabled projects provide a useful comparison always in the work by Annunziata et al. [3]. The co-occurrence patterns observed in quantum repositories—such as RS–BCE (POR 8.750) and SD–OSE (POR 5.333)—were notably stronger than those found in ML projects, suggesting tighter coupling between specific communication and coordination issues in the quantum domain. While ML repositories also exhibit elevated PORs for pairs like OSE–PDE or BCE–OSE, these associations tend to be more evenly distributed and of lower magnitude. Moreover,

the greater number of smell pairs with POR equal to zero in quantum repositories—e.g., PDE–BCE, TF–PDE, and TC–SV—points to a more fragmented or modular manifestation of smells, in contrast to the more interconnected landscape of ML systems. Finally, negative associations were more frequent in quantum projects, indicating divergent team dynamics or counterbalancing patterns that are less apparent in ML contexts. These differences likely reflect the early-stage and specialized nature of quantum software development.

Also here we can provide some implications:

- For *researchers*, the results highlight the need to explore how specific smells interact and whether they trigger or reinforce one another. Strong associations—such as RS–BCE—may indicate cascading coordination failures, while the absence of overlap between smells like PDE–BCE suggests contextual or structural separation. Compared to ML-enabled systems, the sharper and more polarized correlations in quantum repositories call for domain-sensitive models that move beyond treating smells in isolation and toward understanding their interdependencies.
- For *open-source contributors*, these patterns suggest that addressing certain high-risk combinations—such as RS and BCE—may reduce the impact of multiple smells simultaneously. The presence of negative correlations, such as TC–OSE, implies that mitigating one issue may lower the risk of others. Unlike the more entangled smell networks in ML projects, quantum repositories may benefit from focused interventions targeting the most disruptive pairs.

7 Conclusion

The findings of our study revealed not only the widespread presence of community smells but also significant correlations between different smells. As called for in the introduction, this research represents a preliminary yet foundational step toward a more comprehensive investigation of socio-technical issues in the QSE domain. In terms of contributions, we provided:

1. an empirical investigation of the presence of ten community smells in open-source quantum communities;
2. a statistical analysis of the correlations between pairs of these ten smells within the same communities; and
3. a publicly available online appendix [18], provided to ensure replicability and reliability of the findings.

This study opens promising avenues for future research. A more detailed investigation into the specific socio-technical dynamics that lead to the emergence of these community smells is necessary. Such studies could employ qualitative methods to further explore the circumstances surrounding these smells and to identify effective mitigation strategies. Moreover, understanding these socio-technical issues can inform the technical side of software development. Previous

research has suggested a link between community smells and technical debt, and future work should aim to confirm this relationship within the quantum software context.

Acknowledgments. This work has been partially supported by the project 'QUASAR: QUAntum software engineering for Secure, Affordable, and Reliable systems, grant 2022T2E39C, under the PRIN 2022 MUR program funded by the EU NextGenerationEU.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Almarimi, N., Ouni, A., Chouchen, M., Mkaouer, M.W.: 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. pp. 1560–1564. ESEC/FSE 2021, Association for Computing Machinery, New York, NY, USA (Aug 2021). https://doi.org/10.1145/3468264.3473121
2. Almarimi, N., Ouni, A., Mkaouer, M.W.: Learning to detect community smells in open source software projects. Knowledge-Based Systems **204**, 106201 (Sep 2020). https://doi.org/10.1016/j.knosys.2020.106201
3. Annunziata, G., Lambiase, S., Palomba, F., Catolino, G., Ferrucci, F.: How do communities of ml-enabled systems smell? a cross-sectional study on the prevalence of community smells. In: Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering. p. to be defined (2025)
4. Brooks Jr, F.P.: The mythical man-month: essays on software engineering. Pearson Education (1995)
5. Caballero-Espinosa, E., Carver, J.C., Stowers, K.: Community smells—the sources of social debt: A systematic literature review. Information and Software Technology **153**, 107078 (2023)
6. Catolino, G., Palomba, F., Tamburri, D.A., Serebrenik, A., Ferrucci, F.: 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). pp. 11–20. IEEE (2019)
7. De Stefano, M.: An empirical study on the current adoption of quantum programming | Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings
8. De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., De Lucia, A.: Software engineering for quantum programming: How far are we? Journal of Systems and Software **190**, 111326 (Aug 2022). https://doi.org/10.1016/j.jss.2022.111326
9. De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., De Lucia, A.: The quantum frontier of software engineering: A systematic mapping study. Information and Software Technology p. 107525 (2024)
10. De Stefano, M., Pecorelli, F., Palomba, F., Taibi, D., Di Nucci, D., De Lucia, A.: Quantum software engineering issues and challenges: Insights from practitioners. In: Quantum Software: Aspects of Theory and System Design, pp. 337–355. Springer (2024)

11. Gordis, L.: Epidemiology E-Book. Elsevier Health Sciences (Nov 2013)
12. Gote, C., Perri, V., Zingg, C., Casiraghi, G., Arzig, C., von Gernler, A., Schweitzer, F., Scholtes, I.: Locating community smells in software development processes using higher-order network centralities. Social Network Analysis and Mining **13**(1), 129 (2023)
13. Hoare, T., Milner, R.: Grand challenges for computing research. The Computer Journal **48**(1), 49–52 (2005)
14. Hoda, R.: Socio-technical grounded theory for software engineering. IEEE Transactions on Software Engineering **48**(10), 3808–3832 (2021)
15. Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D.M., Damian, D.: The promises and perils of mining GitHub. In: Proceedings of the 11th Working Conference on Mining Software Repositories. pp. 92–101. MSR 2014, Association for Computing Machinery, New York, NY, USA (May 2014). https://doi.org/10.1145/2597073.2597074
16. Knight, W.: Serious quantum computers are finally here. what are we going to do with them. MIT Technology Review. Retrieved on October **30**, 2018 (2018)
17. Lambiase, S., Catolino, G., Tamburri, D.A., Serebrenik, A., Palomba, F., Ferrucci, F.: 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. pp. 67–78. ICSE-SEIS '22, Association for Computing Machinery, New York, NY, USA (Oct 2022). https://doi.org/10.1145/3510458.3513015
18. Lambiase, S., De Stefano, M., Palomba, F., Ferrucci, F., De Lucia, A.: Socio-technical well-being of quantum software communities: A preliminary overview—online appendix. https://doi.org/10.6084/m9.figshare.28307573
19. Moguel, E., Berrocal, J., García-Alonso, J., Murillo, J.M.: A roadmap for quantum software engineering: Applying the lessons learned from the classics. In: Q-SET@ QCE. pp. 5–13 (2020)
20. Palomba, F., Andrew Tamburri, D., Arcelli Fontana, F., Oliveto, R., Zaidman, A., Serebrenik, A.: Beyond Technical Aspects: How Do Community Smells Influence the Intensity of Code Smells? IEEE Transactions on Software Engineering **47**(1), 108–129 (Jan 2021). https://doi.org/10.1109/TSE.2018.2883603
21. Palomba, F., Tamburri, D.A.: Predicting the emergence of community smells using socio-technical metrics: A machine-learning approach. Journal of Systems and Software **171**, 110847 (Jan 2021). https://doi.org/10.1016/j.jss.2020.110847
22. Paradis, C., Kazman, R., Tamburri, D.: Analyzing the tower of babel with kaiaulu. Journal of Systems and Software **210**, 111967 (2024)
23. Piattini, M., Peterssen, G., Pérez-Castillo, R.: Quantum computing: A new software engineering golden age. ACM SIGSOFT Software Engineering Notes **45**(3), 12–14 (2021)
24. Piattini, M., Peterssen, G., Pérez-Castillo, R., Hevia, J.L., Serrano, M.A., Hernández, G., De Guzmán, I.G.R., Paradela, C.A., Polo, M., Murina, E., et al.: The talavera manifesto for quantum software engineering and programming. In: QANSWER. pp. 1–5 (2020)
25. Piattini, M., Serrano, M., Perez-Castillo, R., Petersen, G., Hevia, J.L.: Toward a quantum software engineering. IT Professional **23**(1), 62–66 (2021)
26. Quantum Open Software Foundation: Awesome quantum software. https://github.com/qosf/awesome-quantum-software (2024), accessed: 2024-10-14
27. Quantum Open Software Foundation: Quantum open software foundation (qosf) (2024), https://qosf.org, accessed: 2024-10-14

28. Ralph, P., Chiasson, M., Kelley, H.: Social theory for software engineering research. In: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering. EASE '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2915970.2915998, https://doi.org/10.1145/2915970.2915998
29. Saarimäki, N., Lenarduzzi, V., Vegas, S., Juristo, N., Taibi, D.: Cohort Studies in Software Engineering: A Vision of the Future (Sep 2020). https://doi.org/10.1145/3382494.3422160
30. Saarimäki, N., Moreschini, S., Lomio, F., Penaloza, R., Lenarduzzi, V.: Towards a Robust Approach to Analyze Time-Dependent Data in Software Engineering. In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 36–40 (Mar 2022). https://doi.org/10.1109/SANER53432.2022.00015
31. Shaydulin, R., Thomas, C., Rodeghero, P.: Making quantum computing open: Lessons from open source projects. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. pp. 451–455 (2020)
32. Tamburri, D.A.: Software architecture social debt: Managing the incommunicability factor. IEEE Transactions on Computational Social Systems **6**(1), 20–37 (2019)
33. Tamburri, D.A., Palomba, F., Kazman, R.: Exploring Community Smells in Open-Source: An Automated Approach. IEEE Transactions on Software Engineering **47**(3), 630–652 (Mar 2021). https://doi.org/10.1109/TSE.2019.2901490
34. Tamburri, D.A., Kazman, R., Fahimi, H.: The architect's role in community shepherding. IEEE Software **33**(6), 70–79 (2016)
35. Tamburri, D.A., Kruchten, P., Lago, P., van Vliet, H.: Social debt in software engineering: Insights from industry. Journal of Internet Services and Applications (2015). https://doi.org/10.1186/s13174-015-0024-6
36. Tamburri, D.A., Lago, P., Vliet, H.v.: Organizational social structures for software engineering. ACM Computing Surveys (CSUR) **46**(1), 1–35 (2013)
37. Tamburri, D.A., Palomba, F., Kazman, R.: Exploring community smells in open-source: An automated approach. IEEE Transactions on software Engineering (2019)
38. Tillman, R.E., Eberhardt, F.: Learning Causal Structure from Multiple Datasets with Similar Variable Sets. Behaviormetrika **41**(1), 41–64 (Jan 2014). https://doi.org/10.2333/bhmk.41.41
39. Unitary Fund: Quantum open source software survey. https://unitaryfund.github.io/survey-website/ (2024), accessed: 2024-10-14
40. Voria, G., Pentangelo, V., Della Porta, A., Lambiase, S., Catolino, G., Palomba, F., Ferrucci, F.: Community smell detection and refactoring in slack: The cadocs project. In: 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME). pp. 469–473. IEEE (2022)
41. Wang, X., Cheng, Z.: Cross-Sectional Studies: Strengths, Weaknesses, and Recommendations. CHEST **158**(1), S65–S71 (Jul 2020). https://doi.org/10.1016/j.chest.2020.03.012
42. Yarkoni, S., Raponi, E., Bäck, T., Schmitt, S.: Quantum annealing for industry applications: Introduction and review. Reports on Progress in Physics **85**(10), 104001 (2022)
43. Zhang, Y., Ni, Q.: Recent advances in quantum machine learning. Quantum Engineering **2**(1), e34 (2020)