Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH PAPER

A Novel, Tool-Supported Catalog of Community Smell Symptoms

Antonio Della Porta[1] | **Stefano Lambiase**[1] | **Gemma Catolino**[1] | **Filomena Ferrucci**[1] | **Fabio Palomba**[1]

[1]Software Engineering (SeSa) Lab, Department of Computer Science, University of Salerno, Italy

Correspondence
Corresponding author Antonio Della Porta
Email: adellaporta@unisa.it

Abstract

Software development is a multifaceted endeavor, requiring a profound grasp of both social dynamics and technical intricacies. Poor collaboration often leads to the accumulation of *social debt*, manifesting as unforeseen project costs due to sub-optimal team interactions. *Community smells* have emerged as indicators of these socio-technical inefficiencies and potential social debt. While previous research has focused on automated detection of community smells through analyzing developer communication patterns, our study offers a complementary approach. We emphasize the critical role of project managers in assessing socio-technical dynamics and propose a novel, tool-supported catalog of symptoms. This catalog can be used for manual inspections to identify early signs of community smells at the individual level, allowing managers to address issues before they escalate. Using a mixed-method design that leveraged an existing literature review and a user survey, we cataloged symptoms related to four community smell types. Additionally, we developed `TOAST`, a tool that operationalizes this catalog, and assessed its usability and practical usefulness through an experiment involving project managers. The study showed that even participants unfamiliar with the term "community smells" were able to interpret the tool's output, reflect on team dynamics, and recognize problematic behavioral patterns when supported by structured symptom-based information. The paper concludes by shedding light on the potential impact of our work and its contribution to advancing the detection and analysis of community smells.

KEYWORDS
Social Debt; Community Smells; Mixed-Method Research; Recommendation System.

1 | INTRODUCTION

Software development is a socio-technical activity in which social phenomena, e.g., collaboration and communication between team members, and technical aspects, like the software product or technology adopted, are profoundly connected[1,2,3,4]. Because of this interconnected nature, practitioners need ad hoc artifacts and constructs to navigate the complexity of the development process and achieve the best product. This is even more critical during software maintenance and evolution activities, which often require maintainers to communicate and collaborate (the social) to prevent the inevitable deterioration of a software product (the technical)[5,6]. As teams navigate the complex tasks of updating, refining, and extending software systems, the quality of their interactions significantly impacts the efficiency and effectiveness of their work.

Recalling the famous quote from Tom DeMarco, *"You can't control what you can't measure"*[7], the research community started proposing a plethora of socio-technical metrics, e.g., socio-technical congruence and turnover—resulting in the creation of the so-called *"Social Debt"*, i.e., the unforeseen project costs connected to the presence of poor collaboration and communication conditions within a software community[8]. Soon after, *community smells*—inspired by the well-known Code Smells[9]—arose to characterize sub-optimal socio-technical phenomena in managing a software community that are precursors of Social Debt[8,10]. Since the definition of community smells, the research community started investigating their role in terms of impact[11,12,13,14], the way to predict them[15,16,17,18], and the development of tools for their detection and management[19,20].

While significant progress has been made in understanding and detecting community smells, most existing research has focused on developing automated tools that rely on mining software repositories to identify these socio-technical inefficiencies [11,12,13,14,15,16,17,18,19,20]. These tools are primarily designed to analyze large-scale patterns in communication and code repositories, often producing binary classifications of the presence or absence of a smell based on aggregate indicators. While effective for mining socio-technical trends, such results may lack interpretability for project managers, who must understand the concrete, everyday behaviors behind those classifications to take informed action. Moreover, these automated approaches may overlook subtle, individual behaviors, such as reluctance to collaborate or ineffective communication, that contribute to the emergence of community smells. As a consequence, we argue that by complementing automated tools with a human-centric approach, project managers can achieve a more comprehensive and actionable understanding of the socio-technical environment. This enables the early detection and interpretation of community smells that might otherwise go unnoticed or be misinterpreted in aggregate results. Based on the considerations above, **this paper introduces a complementary approach to community smell detection, emphasizing the critical role of project managers in assessing the socio-technical dynamics within a software community**. Specifically, we propose a novel, tool-supported catalog of symptoms that indicate the emergence of community smells. This catalog is designed to be used by project managers to monitor and evaluate individual developers' behaviors, enabling them to identify and address early signs of community smells before they escalate into more significant issues. This view is supported by a key consideration. Project managers may and may not be familiar with the specific term "community smells", yet many are already sensitive to the underlying behavioral patterns these smells describe, such as poor communication, lack of transparency, or resistance to collaboration, as these are typically extensively discussed in management and organizational behavior literature [21,22]. Therefore, surfacing concrete, observable symptoms may be more effective and relatable for managers than introducing abstract taxonomies or unfamiliar jargon. The symptom catalog is intended to raise awareness by helping managers reflect on and interpret behaviors that align closely with their everyday experience. Unlike binary detection tools, our symptom-based approach is designed to support interpretation, encourage dialogue, and promote the early recognition of problematic dynamics through behavior-oriented cues. In other terms, presenting symptoms, rather than classifications, can thus improve both the accessibility and the practical relevance of community smell detection in real-world contexts.

◎ Research Objective

The objective of this work is to advance the detection and interpretation of community smells by identifying and validating a set of concrete, observable symptoms. By making these symptoms accessible through a tool-supported catalog, the goal is to empower project managers to recognize early signs of socio-technical issues and take informed, proactive action.

From a methodological standpoint, we build the catalog by conducting a mixed-method research study, combining a literature review with empirical data gathered from a survey of experienced practitioners. In addition, we developed a preliminary recommendation system, `TOAST`, in the form of a Discord bot, which allows users to apply the knowledge from our research to effectively manage social debt within their communities. To further assess the tool's practical value, we conducted an experiment involving project managers. Despite their initial unfamiliarity with the terminology behind community smells, participants were able to interpret the tool's output and critically reflect on their team dynamics. Specifically, we recorded 23 true positives, 4 false positives, 21 false negatives, and 37 true negatives. These results yield a precision of 0.88 and a recall of 0.53. In other words, in most cases where `TOAST` flagged a community smell, the participant had independently recognized a related issue: this suggests a strong degree of reliability in the tool's positive detections. This finding supports our hypothesis that presenting concrete, behavior-oriented symptoms can effectively raise awareness and facilitate early recognition of problematic social patterns. To sum up, this work makes the following contributions to the current state of the art:

- A detailed analysis of the symptoms associated with four specific community smells, accompanied by a catalog and an assessment of the importance of each symptom;
- A recommendation system[†] designed to be user-friendly for practitioners, making the research findings accessible and applicable for managing social debt within communities;
- A publicly available online appendix [23] containing all data, scripts, and additional analyses conducted in the scope of our research, to ensure reliability and promote open-source collaboration.

[†] Tool repository link: https://github.com/atdepo/toast-tool

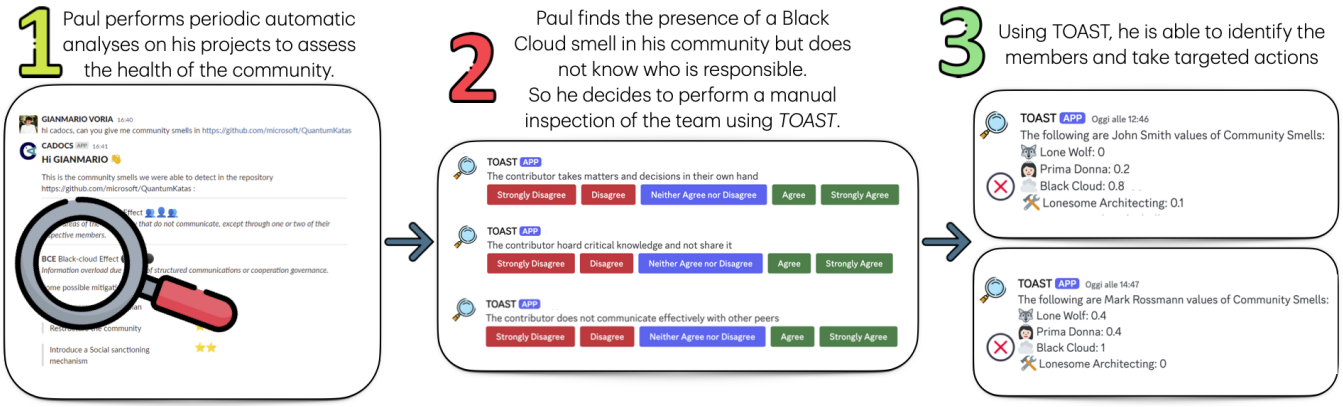

FIGURE 1 Illustrative scenario motivating the use of TOAST. This example highlights the kind of team dynamics the tool is designed to help managers detect and reflect upon.

1.1 | Visionary Scenario

To better illustrate how our approach can be applied, let us consider a visionary scenario that demonstrates the integration of automatic and manual methods for the analysis of the community. This scenario is visually represented in Figure 1 and serves as a motivating example to illustrate the type of situation TOAST is designed to support. Imagine Paul, a seasoned Project Manager, overseeing multiple software projects and teams. His responsibilities include conducting semi-annual team evaluations, managing project timelines, and ensuring the financial health of each project. To maintain efficiency and prevent unforeseen costs, Paul typically relies on automated tools to analyze project repositories and monitor team interactions, identifying potential community smells that could disrupt progress. However, Paul understands that automated tools, while powerful, can sometimes miss the subtle, human-centric behaviors that contribute to socio-technical inefficiencies. For this reason, Paul turns to TOAST, a tool we propose in this study, to assess individual contributors manually. TOAST allows Paul to identify the nuanced behaviors—such as ineffective communication or reluctance to collaborate—that might be overlooked by automated systems. For instance, when TOAST flags a potential *Black Cloud* smell, indicating that critical knowledge is being hoarded, Paul can directly pinpoint the team members who might not be sharing information effectively. The tool provides him with a detailed analysis of communication patterns, recurring behaviors, and individual tendencies that contribute to the smell. Conversely, Paul could begin his analysis by using TOAST during his routine team evaluations. By manually identifying early signs of community smells through the behaviors of individual team members, Paul gains a deeper understanding of the socio-technical dynamics at play. Using these insights, he can then apply automated tools to validate these findings at a broader scale, ensuring that the issues identified are not only isolated incidents but are reflective of deeper, systemic problems within the team or project. This dual approach allows Paul to cross-verify and refine his strategies, addressing issues with a level of precision that significantly reduces the risk of social debt.

By integrating manual assessments with automated detection—or vice versa—Paul can take targeted actions to address issues before they escalate. Whether it is arranging a one-on-one meeting with a specific contributor to discuss their communication habits or setting up a team workshop to enhance knowledge-sharing practices, Paul's proactive and informed approach ensures that his teams remain collaborative, efficient, and aligned with the project's goals. In Figure 1, it is shown how TOAST can help Paul in his activities. In essence, TOAST—and the catalog of symptoms—can empower a manager to move beyond surface-level insights, enabling him to manage the socio-technical dynamics of his teams with unprecedented precision. This integration of manual and automated analysis not only has the potential to enhance practitioner's ability to maintain a healthy project environment but also sets a new standard for community smell detection and management in software development.

The paper is structured as follows: in Section 2, we describe the state of the art of community smells and their detection. Section 3 outlines our research design and methodology. Section 4 presents our research findings, while Section 5 discusses a recommendation system built on the main reserach work findings. Section 6 examines the impact, implications, and validity threats of the research. Finally, Section 7 concludes our study by summarizing findings and proposing future work.

TABLE 1 An overview of the community smells defined in literature, according to Caballero-Espinosa et al. [10].

Community Smells				
1. Architecture by osmosis	7. Cookbook development*	13. Informality excess	19. Newbie free-riding*	25. Prima donnas*
2. Architecture hood	8. DevOps clash	14. Institutional isomorphism	20. Obfuscated architecting	26. Radio silence
3. Black cloud*	9. Disengagement*	15. Invisible architecting	21. Organizational silo*	27. Sharing villainy
4. Class cognition	10. Dispersion	16. Leftover techie	22. Organizational skirmish	28. Solution defiance
5. Code red	11. Dissensus	17. Lone wolf*	23. Power distance	29. Time warp
6. Cognitive distance*	12. Hyper community	18. Lonesome architecting*	24. Priggish members*	30. Unlearning

2 | BACKGROUND AND RELATED WORK

Software development and its engineering are socio-technical activities. To measure the impact of social phenomena on software development, researchers—inspired by the well-known concept of Technical Debt [9]—defined *Social Debt*, i.e., the unforeseen project costs derived from sub-optimal choices in the collaboration and communication aspects—so, management—of a software development team [24]. Moreover, aiming to provide a way to better characterize and identify the source of Social Debt, Tamburri et al. defined *community smell*, i.e., socio-technical anti-patterns whose existence could lead to Social Debt [8].

The research investigated how community smell relates to different aspects of software development. Interestingly, Palomba et al. [11] studied the relationship between community smells and code smells, their product-oriented counterpart, demonstrating that the first are among the top factors influencing the emergence of the last. Furthermore, other researchers focused on establishing the impact of community smells on other dimensions of software engineering, e.g., architecture debt [12] and organizational structure types [13]. Tamburri et al. [14] conducted a large-scale investigation on 60 open-source ecosystems to evaluate (1) community smell diffusion and (2) the perceived impact by developers, showing that not only are smells largely present but they are also perceived to have an impact in the evolution and sustainability of software communities. Such a result shows that socio-technical antipatterns could influence maintenance and evolution in two parallel ways: directly increasing costs (i.e., social debt) and impacting product factors, thus increasing technical-related costs (i.e., technical debt).

More related to their detection, Palomba and Tamburri [17] provided a machine-learning approach to predict community smells considering socio-technical metrics, obtaining promising results (i.e., F-measure of 78%). Moreover, Almarimi et al. [18] proposed a multi-label learning model based on genetic algorithms to detect ten community smells. The work of Almarimi resulted then in the publication of CSDETECTOR [20], a tool implementing the detection strategy described in previous work. Based on the work of Almarimi et al. [20], Voria et al. [19] developed CADOCS, a conversational agent able to detect, given a software repository, ten community smells and propose strategies to refactor some of them. As a final note, Catolino et al. [15] and Lambiase et al. [16] conducted two mining studies, revealing that a correlation seems to exist between gender (in the former) and cultural (in the latter) heterogeneity and the emergence of community smells.

To provide a comprehensive catalog of the community smells defined in literature, Caballero-Espinosa et al. conducted a literature review [10] to identify and catalog the various community smells reported in software engineering research. Table 1 lists the 30 distinct community smells identified in the literature. We then marked with an * the smells that can be identified at the contributor level rather than at community level. These findings have served as the foundation for our research, providing the basis for our empirical study.

When comparing our work with previous research efforts in the field, our study introduces a novel perspective by focusing on the proactive role of project managers in detecting and mitigating community smells. Unlike previous studies that primarily rely on automated detection methods and machine-learning techniques to identify community smells based on socio-technical metrics, our research emphasizes the development of a comprehensive, tool-supported catalog of symptoms specifically tailored for manual inspection by project managers. This human-centric approach not only supplements automated methods but also enables early identification of potential issues by leveraging the insights and observations of project leaders, thereby bridging the gap between technical detection and managerial intervention. Moreover, by incorporating a mixed-method research design that includes both literature review and user surveys, our study provides a more holistic understanding of the factors leading to the emergence of community smells, offering actionable guidance that is both technically informed and contextually relevant for practitioners. On the basis of these considerations, the scientific novelty of our work lies in shifting the focus from automated detection to empowering project managers with actionable insights for early intervention, while the technical novelty lies in the

development of TOAST, a specialized tool designed to operationalize the symptom catalog and facilitate the hands-on detection and management of community smells in real-time project settings.

3 | OBJECTIVE AND RESEARCH DESIGN

The *goal* of this study was to introduce a novel approach for the identification and characterization of community smells through the manual recognition of their symptoms. The *purpose* was to equip practitioners—particularly managers and team leaders—with structured knowledge to make more informed decisions, thereby enhancing the likelihood of software project success. The *perspective* encompasses both practitioners and researchers: practitioners are interested in innovative strategies and tools to effectively address communication and collaboration challenges throughout the software development lifecycle, while researchers are focused on advancing the understanding of the symptoms behind community smells, which can inform future studies and lead to improvements in existing community smell detection techniques.

3.1 | Research Questions

To achieve the study's goal, we conducted a mixed-method research investigation, comprising both qualitative data extraction and a survey study. We began by breaking down our primary goal into two specific sub-goals, each associated with a research question. Below, we introduce each research question along with its motivation.

Our first objective was to develop a new approach for recognizing community smells that could be used both in conjunction with state-of-the-art automated detection methods and independently in a manual community inspection process. To this end, we focused on identifying a comprehensive catalog of symptoms (e.g., unusual behaviors, recurring errors, performance drops, communication breakdowns) that are indicative of these smells, leading to the formulation of the following research question:

> ⍰ **RQ$_1$** — *What are the most commonly reported symptoms of community smells?*

After identifying the key behaviors that practitioners associate with community smells, we sought to further refine and characterize these symptoms to provide more practical guidance for practitioners. This led us to our second research question, which aims to assess the relevance of these symptoms by ranking them according to practitioners' experiences and perceptions.

> ⍰ **RQ$_2$** — *How indicative are the identified symptoms of the presence of community smells?*

Altogether, our study results in a catalog of symptoms associated with community smells, which forms the foundation for a practical diagnostic tool and enhances automated detection methods. The first research question identifies the key symptoms, while the second evaluates their relevance, ensuring the catalog is both comprehensive and practical for real-world application.

3.2 | Research Method Overview

To address our research questions, we employed a three-step methodological approach illustrated in Figure 2:

1. **Participant Selection through Exploratory Survey.** We first conducted an exploratory survey aimed at identifying practitioners with substantial expertise in software development and project management. This preliminary step was required to recruit participants who may have provided credible and insightful data.
2. **Literature Analysis for Initial Symptom Compilation.** Next, we performed a comprehensive analysis of existing literature on community smells, building upon the systematic review by Caballero-Espinosa et al.[10]. This step enabled us to extract an initial set of symptoms associated with various community smells, providing a well-founded starting point for our investigation. Additionally, this analysis allowed us to contextualize our research within the existing body of knowledge, identifying a reasonable set of community smells to focus on.

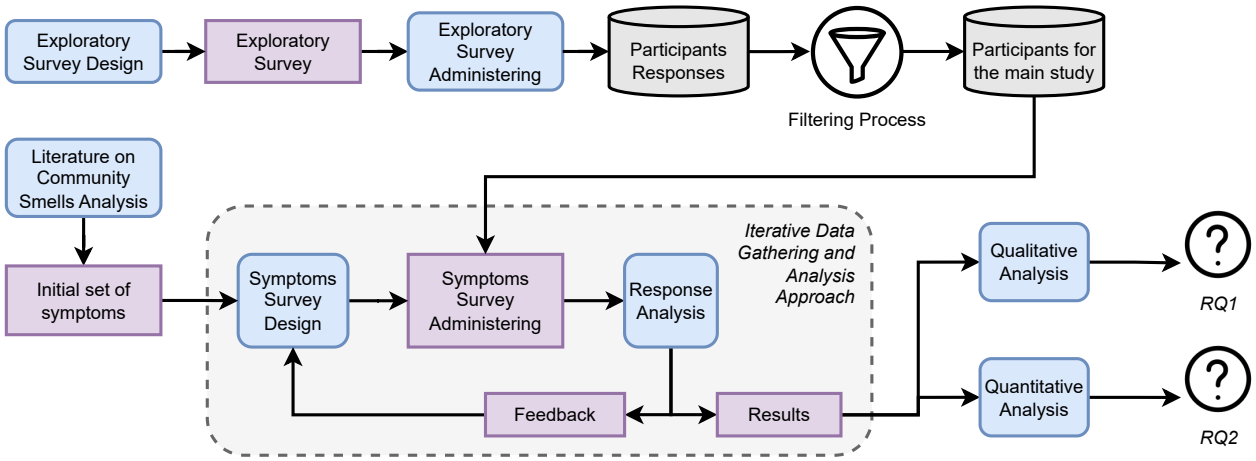

FIGURE 2 Research Method Overview.

3. **Iterative Survey Study for Symptom Refinement and Ranking.** Finally, we conducted an iterative survey study designed with both open-ended and close-ended questions. This dual-format approach qualitatively addressed $\mathbf{RQ}_1$ by capturing insights and experiences from practitioners, while also allowing for quantitative analysis to address $\mathbf{RQ}_2$ through the ranking and evaluation of identified symptoms. The iterative nature of the survey ensured refinement and validation of the data collected, leading to robust and actionable results.

The combination of the three methodological steps is directly aligned with the objectives of our work. The symptoms behind community smells can be accurately identified only through the firsthand experiences of project managers, who possess the deep understanding and practical insights necessary to recognize these complex socio-technical issues within their teams. As such, a survey study represented the most valid research method to employ. Unlike interviews, which are time-consuming and limit the number of participants, surveys allow us to reach a broader, more diverse group of practitioners, enabling the collection of a wide range of experiences and perceptions. Additional quantitative research methods, like statistical analysis or data mining, are not suitable in this context because they rely on existing datasets, which often focus solely on developer interactions within software repositories. These methods may overlook the symptoms of community smells that emerge from the broader social and organizational dynamics best captured through the firsthand experiences of project managers.

Regarding the specific steps taken, the exploratory survey enabled us to engage with knowledgeable practitioners who could provide valuable insights into community smells. The literature analysis established a theoretical foundation for identifying an initial catalog of symptoms. The iterative survey was then instrumental in not only prioritizing these symptoms but also in uncovering additional symptoms based on the practical, real-world experiences of these experts.

For both surveys, we relied on PROLIFIC,[‡] a web-based platform designed for researchers to efficiently sample participants and administer questionnaires. PROLIFIC allowed us to precisely target participants who met our selection criteria by customizing survey preferences and constraints (Section 3.2). Additionally, the platform's reliability metric enabled us to select participants with a proven track record of providing valuable responses, thereby enhancing data quality. It is worth noting that PROLIFIC employs an *opt-in* strategy, which may introduce self-selection or voluntary response bias. To mitigate this potential bias, we adhered to the guidelines proposed by Reid et al.[25], which outline best practices for conducting surveys in the software engineering domain using this platform.

3.3 | Participants Selection—Exploratory Survey

Our first step involved identifying our participant sample; we conducted an exploratory survey to establish a reliable sample of participants for the next stage of the research. To achieve this, we developed an exploratory survey to gauge the managers'

[‡] PROLIFIC website: https://www.prolific.com/

TABLE 2 Exploratory Survey Questions.

ID	Question
S1-1	Nationality
S1-2	What role best describes your current job?
S1-3	What is your company size?
S1-4	Are you familiar with managing distributed teams (where team members are spread across different parts of the globe)?
S1-5	For how many years have you covered a position of team manager or similar?
S1-6	How do you evaluate your experience in Team Management?
S1-7	What is your team size?
S1-8	Have you ever achieved PMI certification? Alternatively, indicate which other certification you have earned.
S1-9	Are you familiar with the concept of Community Smell, i.e., social anti-pattern characterizing communication and collaboration patterns in a development community?
S1-10	If yes, tell us briefly about your experience in that matter

perception of the community smells. Using the data gathered by this survey, we also aimed to gain insight into the state of technology transfers on this matter. It is important to note that since completing this questionnaire, participants have been informed—thus asked to express their agreement—of the possibility of participating in a follow-up survey study.

To design the first draft of the exploratory survey, we relied on the well-known guidelines developed by Kitchenham and Pfleeger[26], along with Andrews et al.[27], which are widely recognized in software engineering research. After, we conducted a pilot study with three practitioners and two researchers in the computer science domain to (1) collect feedback and consequently improve the survey and (2) estimate the time to completion.[§]. Then, we identified the target population of our survey and administered it to them; for the exploratory survey, we set the only criterion as being actively involved in software development as a team member or manager (and similar leading figures). Nevertheless, we asked in the questionnaire for the working role to perform a post-execution evaluation. According to Flanigan et al.[28] advice, we ensured the survey remained anonymous to avoid influencing participants' responses. Furthermore, according to similar studies, we included two attention-check questions to ensure data reliability. The survey was created using a Google Form and was designed to be completed within 10 to 15 minutes.

The exploratory survey questionnaire was designed as a mix of open- and close-ended questions. It was composed of four sections. In the first one, we provided participants with general information on the study, data policy, and requests for agreement. The second section asked for general information on the participants, such as nationality, for a demographic analysis of the sample. It is important to note that the platform used for administering the questionnaire (i.e., PROLIFIC) already provided us with such knowledge; we asked them for reliability and attention-check reasons. We also collected the gender of participants for demographic reasons; we did not ask participants directly but used the information they provided in the public participant data (this data is sent to Prolific at the registration stage). The third section asked for working data, e.g., company size, working role position, and self-assessed experience in software development or management. Moreover, we asked participants if they had certification in project management (e.g., PMI[¶]) as a way to assess their skills. Last, in the fourth section, we asked participants about their experiences with community smells (after providing them with the definition) and asked them to provide us with a textual description of these characteristics. Then, we thanked the study participants and provided an opportunity to leave feedback on the questionnaire. Table 2 reports the complete list of questions. All demographic information was collected primarily to characterize our sample and provide more reliability to our results, as well as provide some statistical data to support our results.

After the dissemination of the exploratory questionnaire and the responses given by the participants, we performed a data analysis phase to extract insights and information for the answers. To perform such analysis we used the R programming language with the core libraries to analyze the data collected from the questionnaire. To start our analysis, we selected only the managerial figures from the sample based on the responses given to the *S1-2* question, which defines the job role. The answers provided by the managers were analyzed to help us understand the demographics of our sample, their field expertise, and their awareness and knowledge of community smells, which are essential for assessing the status of technology transfer. In particular, we analyzed the responses to question *S1-9* to estimate the number of managers who had never experienced community smells (value 0 on the Likert Scale) and those who had (values greater than 0). Among the managers who reported experience with

[§] The pilot highlighted an estimated completion time of 2 minutes. Feedback from the pilot participants led us to change the question on community smell knowledge level, i.e. *S1-9*, and add an open question to add personal experiences with the smells, i.e. *S1-10*, to gauge (1) their perception of the topic and (2) the form on which they experienced them

[¶] Project Management Institute website: https://www.pmi.org/ (8 August 2024)

TABLE 3 Inclusion Criteria

ID	Question	Criteria
S1-5	For how many years have you covered a position of team manager or similar?	≥ 3 years
S1-6	How do you evaluate your experience in Team Management?	≥ 4
S1-8	Have you ever achieved PMI certification? Alternatively, indicate which other certification you have earned.	Yes or Other Certifications

community smells, we further examined their responses to question *S1-4*, which asked about prior experience with managing distributed teams, to qualitatively explore potential patterns between team distribution and smell awareness. This was intended as an exploratory cross-check based on survey answers. Additionally, we reviewed answers to the open-ended question S1-10 to gain deeper insight into how managers interpreted and described their experiences with community smells. Through such a mixed analysis, we were able to cluster each participant based on community smell knowledge, thus providing a picture of the overall knowledge of the matter and experience in the software development field.

The answers provided for this questionnaire were mainly used to select the participants' sample for the main survey study (described in Section 3.5) aimed at answering our research question; we selected from the managers of the exploratory survey only the ones that satisfied our criteria, reported in Table 3. Specifically, we evaluated the managers in terms of (1) years of experience, (2) self-assessed evaluation of managerial skills, and (3) obtained certifications.

3.4 | Analysis of the Literature on Community Smells

Since our study involved identifying a set of symptoms of community smells, we started by extracting what was already available in the literature. By doing so, we wanted to (1) be sure to capture already identified patterns to start asking with and (2) identify which community smells best suited our goals.

3.4.1 | Literature and Smells Identification and Selection

We did not perform a literature review ourselves since Caballero-Espinosa et al.[10] published a recent one in *Information and Software Technology*. This review provided a thorough overview of community smells, making it highly relevant to our study's objectives. Thus, instead of duplicating efforts, we analyzed their work to extract the specific information needed for our research.

We then selected the set of community smells that best aligned with our goals by focusing on the type of analysis that would be most beneficial to project managers in the intended use case: diagnosing issues at the level of *individual contributors*. Since our research aims to develop a tool that project managers can use to identify and address community smells, we concentrated on those originating from individual behaviors rather than at the organizational level. This focus is particularly useful because it allows managers to detect early signs of community smells at the contributor level, enabling more precise and timely interventions.

The procedure we envision is especially valuable for this type of analysis because individual-level smells are more directly observable and actionable within a team setting. Managers can more easily recognize and address these symptoms, which often manifest as specific behaviors or patterns within their teams. This targeted approach not only makes the tool more practical for day-to-day use but also ensures that interventions can be made before these issues escalate into larger, more systemic problems. Starting from the 30 community smells identified by Caballero-Espinosa et al.[10], we applied a two-step procedure to identify those that originate from individual behaviors. The catalog by Caballero-Espinosa et al. provides detailed descriptions and contextual information that helped us identify which smells are typically attributable to individual behaviors rather than team-wide or structural issues. Specifically, we focused on those smells whose definitions and documented causes suggest they can be meaningfully recognized by observing the actions, communication styles, or decision-making patterns of individual contributors. This filtering procedure was guided by two steps:

- At first, we reviewed the cataloged descriptions in [10] to identify smells where problematic behaviors are framed around individuals (e.g., developers who act autonomously, resist collaboration, or hoard knowledge). For instance, *Lone Wolf* describes a developer who isolates themselves from the rest of the team, while *Prima Donna* refers to contributors who demand special treatment and disregard team conventions; these both clearly center on individual attitudes and actions, as opposed to smells like *DevOps Clash*, which emerge from structural tensions between development and operations

teams, resulting in slower development cycles and operational inefficiencies. This yielded a preliminary list of ten smells that we considered potentially observable at the contributor level: (1) *Black Cloud*; (2) *Lone Wolf*; (3) *Prima Donna*; (4) *Lonesome Architecting*; (5) *Cookbook Development*; (6) *Disengagement*; (7) *Newbie Free-Riding*; (8) *Cognitive Distance*; (9) *Organizational Silo*; and (10) *Priggish Members*.

- We then cross-checked the corresponding primary studies referenced in the systematic review to confirm that the symptoms associated with each smell could realistically be assessed through individual observation by a project manager. This step ensured that the smells aligned with our goal of developing a human-centric tool (`TOAST`) for manually assessing social dynamics in software teams. This second step confirmed that the ten smells indeed present patterns of behavior that can be recognized through direct, interpersonal interaction, such as communication style, level of engagement, and autonomy, without relying on organizational-level data or repository mining.

Once we had refined the set of candidate community smells, we held an in-depth internal discussion within the research team and ultimately decided to focus on a final set of four smells, whose description is reported in the following:

Black Cloud[24]. This smell occurs when organizations do not provide the conditions for social interactions and effective communication between teammates, thus not supporting the exchange of knowledge during software development processes.

Lone Wolf[14]. This smell occurs when defiant teammates carry out their work irrespective or regardless of their peers, reflecting poor communication addressing project needs. The effects are, for instance, unsanctioned architecture decisions across the development process, code smells, and project delays.

Lonesome Architecting[29]. Non-architect teammates see the need to make architecture decisions because the current architects are too few and far apart. From a social perspective, developers are unaware of what they are doing. Also, this scenario leads to incompatibility problems and faster decision-making.

Prima Donnas[24]. This smell indicates the presence of teammates working in isolation. They are unwilling to welcome the change of legacy products and support from other teammates. These teammates prevent the organization from innovative solutions or processes and effective communication and collaboration.

We recognize that other contributor-level smells could also have been valid choices. However we opted to focus on these four for three main reasons:

- First, they have been widely investigated and discussed in the literature (e.g., [16,11,30,29]), providing a foundation for building and validating the catalog of symptoms. As such, we chose to focus on community smell types that could serve as a starting point and baseline for both prior and subsequent studies on community smells.
- Second, we aimed to keep the study tractable for both researchers and participants. Including a larger number of smells would have substantially increased the length and complexity of the survey instrument described in Section 3.5, potentially leading to fatigue among participants and lower quality responses. This, in turn, could have affected the reliability of the findings and complicated the data analysis. Each additional smell would have required a dedicated set of symptom-related questions, significantly expanding the questionnaire and increasing the burden on both respondents and analysts.
- Finally, as part of our study, we conducted an experiment with project managers to evaluate how they perceived and interpreted the recommendations provided by the tool that implements the symptom catalog (see Section 5.3.2). Given the depth and complexity of this evaluation, we deliberately focused on a subset of community smells to allow for a more focused and interpretable analysis, thus enhancing the reliability and practical relevance of our findings.

As a final remark, it is worth mentioning that our selection was not based on the usefulness of community smells from the perspective of project managers. To the best of our knowledge, there is currently no empirical literature that systematically investigates the relative importance or perceived utility of different community smells in managerial practice. As a result, this criterion would have been difficult to apply in our selection process. Instead, our selection was first guided by a pragmatic criterion: we prioritized smells that have been widely studied and discussed in the literature, under the assumption that sustained scholarly attention serves as a proxy for practical relevance and impact. In this sense, we treated the prominence of smells in prior work as an indirect indicator of their potential usefulness to project managers.

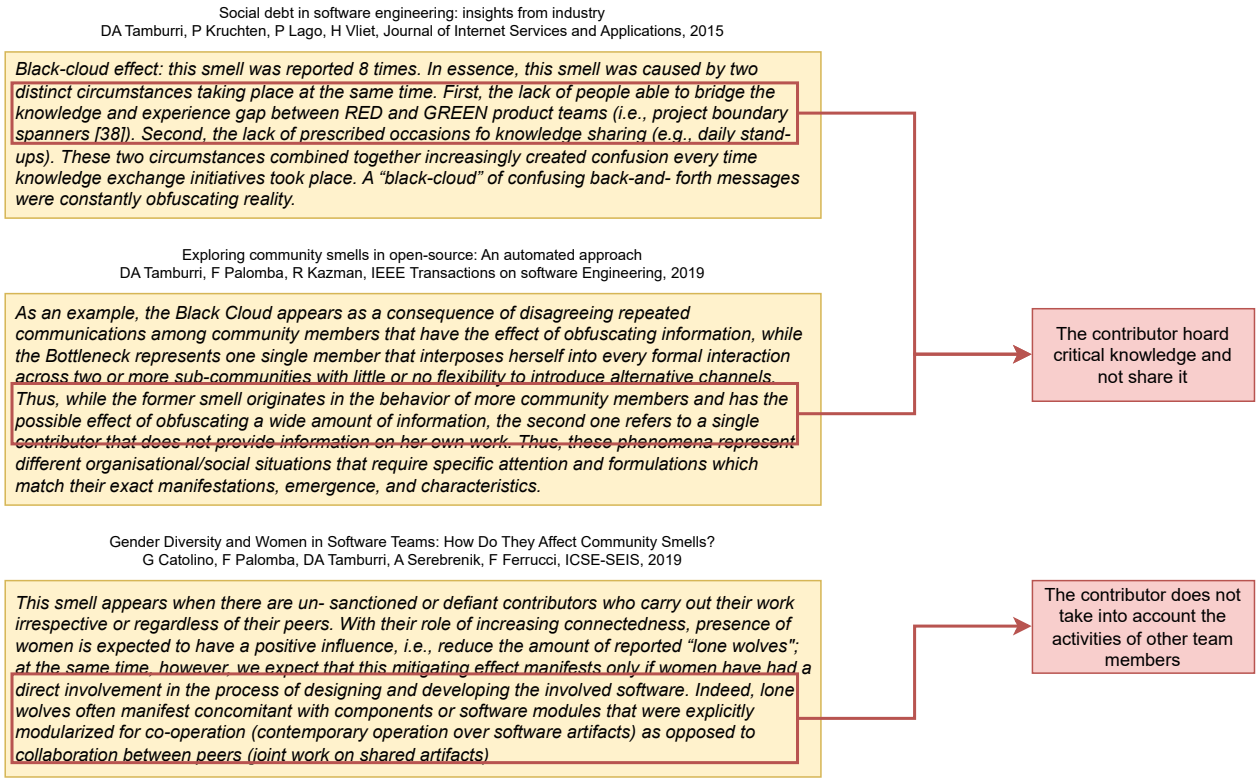

FIGURE 3 Example of the Coding Process.

3.4.2 | Analysis of the Literature

Once the target community smells were selected, we proceeded to identify the symptoms associated with each smell through a *literature-driven* coding process. Specifically, we reviewed all primary studies referenced in the systematic review by Caballero-Espinosa et al.[10], focusing on text segments that described the causes, manifestations, or effects of each smell. To extract and synthesize this information, we applied a round of *deductive structural coding*[31]. This involved categorizing relevant excerpts using a predefined structure composed of two elements: the cause of the smell and the smell itself. Our goal was to formulate concise, behavior-oriented symptom statements that project managers could plausibly observe during daily interactions with their teams (e.g., *"reluctance to interact or share knowledge"*). This coding activity was conducted manually by two authors, with the coded data then independently reviewed and cross-checked by a third author to ensure consistency and reduce subjectivity. In addition, we extended our analysis to other relevant studies on the selected smells beyond those included in the original review, using the same coding approach. Figure 3 provides a visual example of this coding process.

The first author began by thoroughly reviewing all identified literature and systematically extracting the necessary information. This data extraction process was straightforward, with the first author compiling all relevant details—such as the paper, key text, and associated smells—into a structured sheet. The extraction of causes from the identified text was a collaborative effort between the first two authors, allowing for a comprehensive analysis of the data. Given the manageable amount of data, the authors were able to review and discuss the findings together thoroughly. Similar causes were then iteratively consolidated into singular concepts until no further merging was possible. The outcome of this process, presented in our online appendix[23], was a refined set of symptoms for the selected smells, which served as the foundation for developing the questionnaire.

3.5 | Symptoms Survey

To refine and validate the set of symptoms derived from the literature, we adopted an *iterative survey-based approach* involving practicing project managers. This effort also served to address our two research questions: (i) identifying human behaviors that can indicate the presence of community smells, and (ii) evaluating the perceived importance of these symptoms in practice. The study was structured into two rounds of surveys - more details on the execution of these two rounds are reported in Section 4. Each round combined both open- and close-ended questions, allowing for a mix of qualitative and quantitative analysis. Open-ended responses enabled us to apply a qualitative coding process to uncover practitioner-reported symptoms, while close-ended questions were used to quantitatively assess the clarity, observability, and relevance of each symptom. In the first round, a final set of 15 experienced project managers were presented with the symptom set derived from the literature. They were asked to rate each symptom for clarity and practical relevance, and to provide open feedback on any items they found unclear, redundant, or inapplicable. This round also allowed participants to suggest additional symptoms based on their experience. The collected responses were systematically analyzed to refine the initial catalog: we revised phrasing, consolidated overlapping items, and added new symptoms grounded in practitioner input. This ensured that the resulting catalog reflected both theoretical grounding and real-world observability. A second round of validation involved a different group of 10 project managers, who reviewed the revised and expanded symptom catalog. Their task was to reassess the updated symptoms to confirm clarity, relevance, and the absence of redundancy. This second validation phase provided further assurance that the final symptom set was well-grounded, practically useful, and suitable for implementation within our tool.

More specifically, we designed the survey following established guidelines for personal opinion surveys in software engineering[26,27,28]. A pilot study confirmed an estimated completion time of 15 minutes and did not lead to major changes. Attention-check questions were included to ensure data quality. The questionnaire was made up of seven sections. The first section provided an introduction for participants and included details of the study, data policy information, and a request for agreement. The second section contained the general definition of community smells to ensure that participants had the same understanding of the topic. The next four sections were based on the four community smells identified during the literature analysis step (Section 3.4). Each section began with the definition of the smell and followed with questions asking participants how many times they had experienced the smell and if it had a tangible impact on communication and collaboration in their team. Additionally, for each smell, participants were asked to indicate (using a Likert Scale from 1 to 5) their agreement with the relationship between the symptoms identified for the smell and the smell itself. Furthermore, for each smell, an open-ended question was included to gather any additional symptoms experienced by the participant. Lastly, to gain more insight, participants were asked to provide feedback on other anti-patterns not included in the questionnaire and to express any concerns or feedback on the questionnaire itself. The analysis of the open-ended questions was made similarly to the one described in Section 3.4.

To address $\mathbf{RQ}_1$, we synthesized insights from both the literature and the survey responses. Specifically, we used a deductive coding approach to analyze participants' answers to open-ended questions, building upon the initial symptom candidates identified in our literature review. This process enabled us to validate the original symptom set, refine unclear or redundant items, and incorporate new practitioner-suggested symptoms. The outcome of this combined analysis is a **refined** and **practitioner-informed catalog of observable behaviors**, which constitutes our answer to $\mathbf{RQ}_1$.

To address $\mathbf{RQ}_2$, whose aim was to provide a ranking of the symptoms, we conducted a statistical analysis of the participants' close-ended responses. For each candidate behavior, we aggregated the ratings and computed two descriptive statistics: the median, to capture the central tendency of perceived importance, and the standard deviation, to measure the level of agreement among participants. This analysis allowed us to **rank the symptoms based on their perceived relevance and consistency of judgment across respondents**. To make the scores for the symptoms more informative, we weighted each manager's answers by the frequency with which they reported encountering the smell, i.e., using a Likert scale from "never" (weight = 1) to "always" (weight = 5). This frequency score served as the weight when aggregating answers. While this weighting scheme is not derived from prior studies or established guidelines, it rests on a simple premise: managers who have experienced a particular smell more frequently are likely to have deeper and more reliable insights into its behavioral manifestations than those with limited exposure.

3.6 | Ethical Consideration

Studies involving human participants in our country do not require approval from an Ethical Review Board yet. Nevertheless, the survey design considered numerous ethical and privacy concerns. Below, we outline the precautions we have taken to ensure full compliance with ethical considerations:

- All activities related to the surveys were entirely anonymous. We recorded no identifying information of the involved participants.
- We made it clear to participants that they could withdraw their survey submission at any time and that no information entered up to that point would be tracked.
- We followed the indications of PROLIFIC to ensure alignment between privacy policies.

These measures were all agreed upon and explicitly communicated by the authors of the paper. All of them were made clear to the participants before any surveys for our research were conducted.

4 | ANALYSIS OF THE RESULTS

In this section, we highlight the study's main findings and answer the research question posed in the previous chapter.

4.1 | Participant Selection—Exploratory Survey

The first step of our research involved administering the exploratory questionnaire (described in section 3.2) to a set of software engineers using PROLIFIC. With this questionnaire, we primarily aimed to identify reliable managerial figures to collect data for our research questions. We also wanted to find out if managers know about these problems, how often they see them, and what they think about them since this helps us connect our research with what actually happens in software teams.

We received 304 responses from participants across various regions worldwide, as can be seen in Figure 4.

A demographic analysis shows that among them, 78 participants (25.7%) are from the United Kingdom, and 49 (16.2%) are from the United States. Additionally, there is representation from non-Anglophone nations, with 30 (9.90%) from Portugal and 28 (9.24%) from South Africa, among others. The sample exhibits diversity not only geographically but also in participant roles. The most prominent role in the sample is the Project Manager with 96 participants (31.7%), followed by Development Team Member with 76 participants (24.1%), and Software Architect with 30 participants.

In order to understand whether the sample of participants was familiar with the concept of community smells and to measure the state of technology transfer, we leveraged the answers given to the closed-question *S1-9* (Table 2). From the entire set of participants, we found that 245 (80.86%) were not familiar with it, 26 (8.58%) were familiar, and 32 (10.56%) were not sure. This information exposes the fact that from the entire set of participants, the majority of them had never heard of the concept of community smell. Even by selecting only managerial figures, using the answers from question *S1-2* (Table 2) and thus obtaining a subset of 140 managers, the percentages do not vary: 111 of the managers were not familiar with the community smells (79.28%), 16 of them were familiar (11.42%), and 13 were not sure (9.28%). This result is not necessarily surprising. Managers are generally more concerned about the underlying causes and observable symptoms that hinder effective teamwork, which are concerns extensively explored in the management and organizational behavior literature. As such, they may already recognize many of the issues that community smells describe, even if they are unfamiliar with the specific terminology. For instance, prior research on teamwork effectiveness (e.g., [32,33]) highlights recurring challenges such as poor communication, lack of transparency, and resistance to collaboration, all of which closely align with the behaviors captured by community smells. This conceptual overlap reinforces the motivation behind our work. Rather than imposing abstract taxonomies or technical jargon, our approach seeks to raise awareness by helping managers reflect on and interpret concrete, observable patterns that may signal emerging problems within their teams. The goal is not automated labeling, but guided interpretation and early recognition of problematic dynamics. As described later in Section 5.3.2, this positioning is further supported by the findings of the preliminary experiment conducted to assess the practical relevance of our catalog of community smell symptoms: despite being largely unfamiliar with the terminology, the managers who participated to the experiment were able to make sense of the tool's output and reflect critically on their own team contexts. In several cases, the structured symptom descriptions prompted a reassessment of prior assumptions, suggesting that the framework can serve as a catalyst for awareness and more informed managerial action.

To broaden the scope of the discussion, it is worth reporting that, while analyzing the responses to the open-ended question *S1-10*, we realized that the participants predominantly perceive community smells as communication issues. To further explore this perception, we generated a word cloud, shown in Figure 5, from the open-ended responses provided by participants. As shown in the figure, the most prominent terms, such as "communication", "community", "team", and "patterns", indicate a strong association between community smells and communication-related challenges. This visual representation underscores

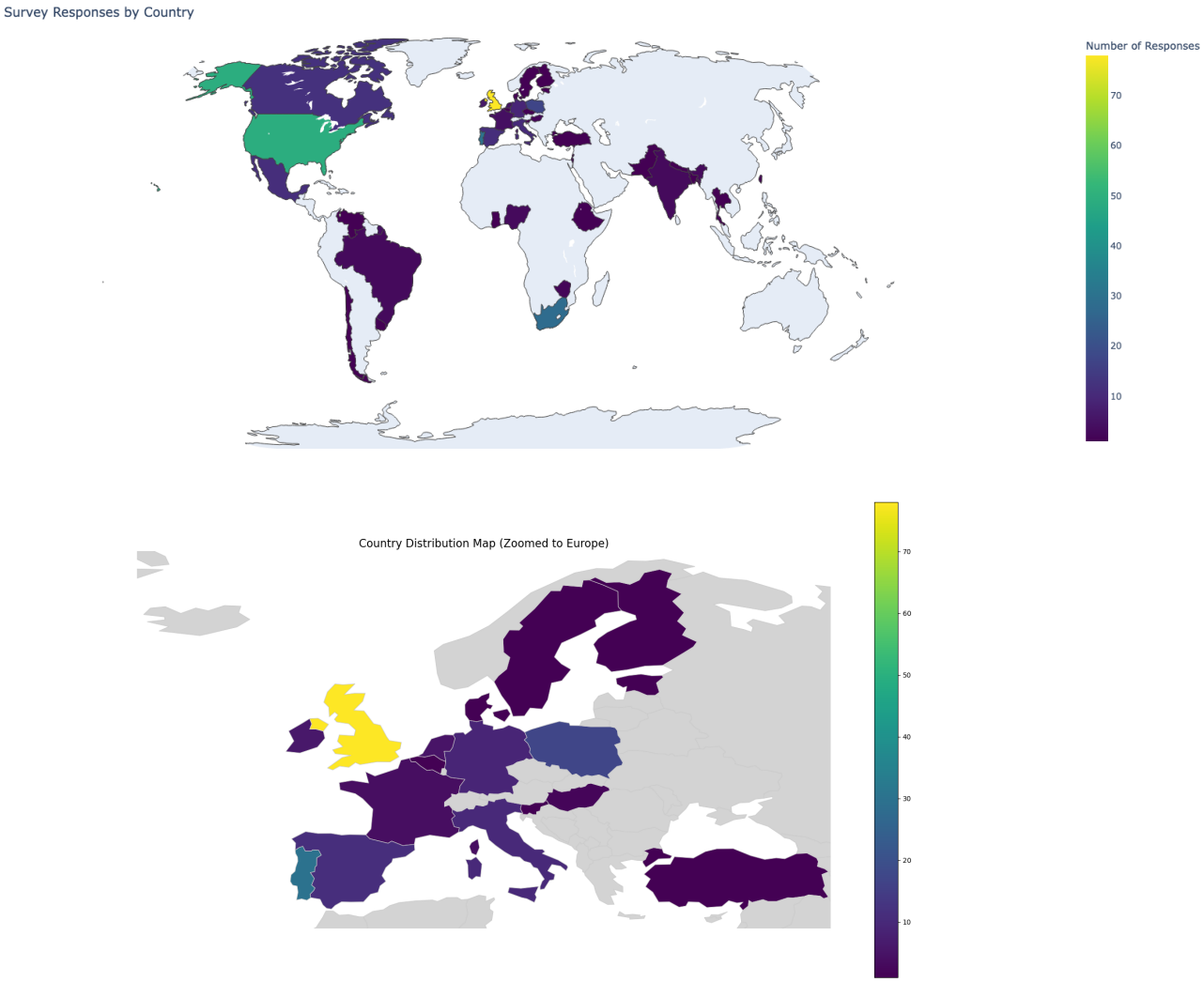

FIGURE 4 Geographical distribution of participants in the Exploratory Study. The top map shows global distribution, while the bottom focuses on the European region.

that, according to participants, ineffective communication is a key factor in the emergence of community smells. The frequency of terms like "collaboration", "development", and "interaction" further emphasizes the critical role of effective communication and teamwork in mitigating these socio-technical issues, and is also in line with the existing body of knowledge in management and organizational science where multiple articles have studied the impact of team communication and composition on effective teamwork[32,33]. This finding has also two significant implications in the software engineering field. Firstly, our results diverge slightly from the existing body of knowledge on community smells, as they suggest a stronger emphasis on communication problems than previously documented, which traditionally focuses on broader socio-technical factors, like collaboration among team members. More importantly, it emphasizes the need for tools that complement automated approaches, as communication issues are inherently tied to behavioral dynamics that influence how developers interact. This reinforces the motivation behind our study, which aims to develop an instrument that enables managers to diagnose individual behavioral symptoms of community smells, like those rooted in communication issues. Following our analysis of the selection survey responses, we developed and applied criteria to identify a subset of expert managers from the exploratory survey participants. This refined group was then targeted for our main survey distribution. The set of inclusion criteria is defined in Table 3 and maps the questions chosen to be the filter of our sample to the inclusion criteria to match in order to be selected. After applying our inclusion criteria to the 303 survey respondents, we identified 31 expert managers who qualified to receive the main questionnaire focused on community smell symptoms that will help us answer the study research questions.

FIGURE 5 Word Cloud Of The Answers Of *S1-10*

4.2 | RQ_1: What are the most commonly reported symptoms of community smells?

The focus of this section is to address $\mathbf{RQ_1}$, namely the identification and validation of observable behavioral symptoms associated with each of the four selected community smells. As discussed earlier, we selected 31 experienced project managers through an initial exploratory survey. These participants were then divided into two subgroups across consecutive iterations of data collection and analysis. This iterative process (detailed in Section 3) served to validate the symptom set derived from the literature and to uncover additional practitioner-reported symptoms, thereby refining the final catalog.

In the first iteration, we distributed the initial survey to 15 participants. The questionnaire included both close- and open-ended questions, allowing participants not only to rate symptoms but also to provide free-text feedback and additional symptom suggestions. The open responses were subjected to qualitative analysis, which surfaced several practitioner-informed refinements. For instance, in response to the Black Cloud section, one participant noted: *"When someone is hoarding critical knowledge and not sharing it or when there is not an effective communication between team members"*. This led us to add a new symptom, i.e., *"The contributor hoards critical knowledge and does not share it"*, to the *Black Cloud* category. Although this behavior could also plausibly relate to other smells such as *Lone Wolf* or *Prima Donna*, we assigned it to *Black Cloud* based on how this smell is typically framed in the literature, namely as involving passive resistance, disengagement, and withdrawal from collaborative activities. It is important to note that such symptom-to-smell mappings reflect the *dominant behavioral framing* rather than *exclusive associations*. Overlaps are not only possible but expected, as different smells may share similar surface behaviors while differing in intent or context. The feedback also helped clarify distinctions between potentially overlapping symptoms. For example, symptoms such as *"unwillingness to accept help or support"* and *"refusing to consider others' ideas or opinions"* were retained as separate items based on how participants described them. The former refers to a developer's resistance to assistance, often stemming from excessive autonomy or overconfidence, while the latter reflects a dismissive attitude during collaborative exchanges (e.g., planning meetings or design reviews). Based on these considerations, we ended up with Symptoms #3 and #4 in Table 4, which reflect these two distinct behavioral patterns. Additionally, while symptoms like *"reluctance to interact or share knowledge"* might intuitively align with both *Lone Wolf* and *Prima Donna*, we assigned it to *Lone Wolf* because this behavior, i.e., deliberate avoidance of communication and team engagement, is its defining characteristic. Although *Prima Donna* may also resist collaboration, such behavior is often secondary to the smell's hallmark traits (e.g., demanding special treatment or breaking team norms).

Following the first iteration, we incorporated validated and newly proposed symptoms into a revised survey. This updated version was distributed to the remaining 16 participants. Eleven completed the questionnaire, and one was excluded due to failing an attention check. This second round of responses confirmed the symptom set and did not yield further additions, at which point we concluded the iterative process.

In total, we identified and validated 10 symptoms: two for *Lone Wolf* (both from literature), two for *Prima Donna* (one from literature and one from practitioner input), three for *Black Cloud* (one from literature and two practitioner-derived), and three for *Lonesome Architecting* (all from literature). Table 4 summarizes the final catalog and sources for each symptom.

More generally, it is important to emphasize that many community smells, particularly those centered on individual behaviors, naturally share common or closely related symptoms. This overlap is not incidental, but rather reflects the inherent nature of social anti-patterns: distinct smells may arise from similar underlying behavioral tendencies or attitudes. For example, both

TABLE 4 Catalog of the symptoms elicited from our work, along with their impact on the emergence of community smells.

ID	Community Smell	Symptom Identified	Source	Weighted Mean	Standard Deviation	Score
1	Lone Wolf	The contributor has insufficient communication with the team	Literature	3.7	1.0	3.8
2	Lone Wolf	The contributor does not take into account the activities of other team members	Literature	4.1	0.9	4.6
3	Prima Donna	The contributor resists receiving help or support from peers, such as mentoring or technical assistance	Literature	4.2	0.9	4.7
4	Prima Donna	The contributor dismisses or refuses to consider others' ideas or opinions during team discussions	Survey	4.0	0.9	4.4
5	Black Cloud	The contributor takes matters and decisions in their own hand	Literature	3.4	1.3	2.7
6	Black Cloud	The contributor hoard critical knowledge and not share it	Survey	3.9	1.1	3.6
7	Black Cloud	The contributor does not communicate effectively with other peers	Survey	4.0	1.0	4.0
8	Lonesome Architecting	The contributor complained of a lack of knowledge of the product requirements	Literature	3.6	1.1	3.3
9	Lonesome Architecting	The contributor complained of a loss of general vision of the product	Literature	3.4	1.2	2.7
10	Lonesome Architecting	The contributor was called upon to make architectural decisions that were not his responsibility	Literature	3.7	1.3	3.0

Lone Wolf and *Prima Donna* involve individualism and resistance to collaboration, which can manifest through comparable observable behaviors. The findings reported for $\mathbf{RQ_1}$ illustrate this dynamic, as several symptoms showed conceptual proximity across different smells. However, our aim was not to artificially disentangle these overlapping traits, but to surface concrete, observable patterns that project managers can use as a structured lens to assess and interpret team dynamics.

4.3 | $\mathbf{RQ_2}$: How indicative are the identified symptoms of the presence of community smells?

Our second question aimed to evaluate the weight of each symptom associated with smell. To achieve this, we quantified the importance of each behavior by means of a summary measure.

First, we leveraged the responses to the questions of the perceived importance of a symptom identified, whose answers were rated on a Likert Scale ranging from 1 to 5. Initially, we conducted a weighted mean computation for each question, integrating the familiarity rating provided in the initial question of each survey section as the weighting factor—this methodological choice aimed to accommodate participants' varying degrees of familiarity with the community smell under scrutiny. The weights range between 0.2, which corresponds to the answer "*Never*" to 1, which corresponds to the answer "*Always*", with incremental steps of 0.2. Then, we computed the weighted means and assayed the standard deviation for each question's responses. This statistical metric enlightened the dispersion or variability characterizing participants' opinions concerning the importance of each symptom. Lastly, we derived a distinct score for each question by computing the weighted mean and standard deviation ratio. This scoring framework enabled the quantification of the symptoms that garnered high scores of perceived importance and manifested relatively low variability in participant responses. So, the higher the score is, the more the community smell to which it refers can be present (according to the managers' perception). This metric, along with all others produced by this statistical analysis, are listed in Table 4.

The analysis revealed key insights. Symptoms with particularly high scores, such as *"The contributor does not take into account the activities of other team members"* and *"The contributor refuses to listen to the ideas or opinions of peers"*, are strong indicators of the presence of community smells like Lone Wolf and Prima Donna. These findings suggest that project managers should prioritize monitoring these behaviors as early warning signs of potential socio-technical issues.

At the same time, symptoms with lower scores, such as *"The contributor takes matters and decisions into their own hands"* and *"The contributor was called upon to make architectural decisions that were not their responsibility"*, while still relevant, may be more context-dependent and less universally recognized as indicators of community smells. This implies that these symptoms might require additional context or corroborating evidence before being flagged as significant concerns.

In conclusion, the metrics produced by this statistical analysis provide a practical diagnostic tool for project managers. By focusing on the highest-scoring symptoms, managers can proactively address potential issues within their teams, thereby contributing to a healthier and more collaborative development environment.

5 | TOAST: A COMMUNITY SMELL SYMPTOMS TRACKING TOOL

To make our research results practical and actionable for practitioners, we developed a simple recommendation system as a proof of concept for leveraging the catalog of symptoms we identified and validated. This tool, named TOAST (**T**eam **O**bservation **A**nd **S**mells **T**racking tool), is built on the Discord[#] platform. TOAST empowers managers to conduct a manual analysis using the catalog of symptoms, drawing on their management expertise and perspective to identify and mitigate community smells early. The complex social dynamics underlying community smells [34,15,29] suggest that relying solely on automated detection tools may not fully capture these phenomena within team structures. The development of TOAST is driven by two key objectives: (1) Enhance practitioners' awareness of potential issues, which is critical for effective problem resolution; and (2) Integrate seamlessly with the widespread use of recommendation systems in software development environments. The source code of the tool can be found in the online appendix [23], complete with the installation steps and all the needed procedures to generate and retrieve the token needed to start the application correctly.

It is worth emphasizing that TOAST is intended to complement, not replace, existing evaluation processes. The final interpretation of its scores should be integrated with other information sources to have a comprehensive understanding of the situation. In this section, we outline the key features of TOAST and report practical use cases that illustrate how the tool can be employed.

5.1 | Tool Functionalities

TOAST employs a survey-based methodology to identify community smells where a manager can fill out a questionnaire based on the catalog of symptoms, aiming at recognizing the presence of symptoms of community smells within the team. Each behavior is presented as a specific statement, and managers are asked for each team member to indicate their level of agreement or disagreement regarding the individual's adherence to these behaviors on a Likert Scale of 5 values ranging from *Strongly Disagree* to *Strongly Agree*.

User interaction with TOAST to analyze a single contributor follows these steps:

Step 1. Interaction is initiated through the `/start` command in the Discord chat interface. Upon activation, the bot presents itself, as can be seen in step 1 of Figure 1, issues a greeting message and presents to the user the three primary functionalities: the option to **start an analysis** of a team contributor or manage team composition through **member addition or removal**.

Step 2. When the user selects the analysis process, the bot systematically presents survey questions one by one, each accompanied by interactive response buttons to facilitate manager input, as can be seen in step 2 of Figure 1. These questions encompass all validated symptoms associated with community smells. As the manager progresses through the survey, responding to each question, the bot records and processes the inputs.

Step 3. Upon answering all the 10 questions TOAST proposes, the analysis process terminates, and a final summary containing all the scores calculated for each community smell relative to the contributor in question is shown as can be seen in step 3 of Figure 1. After the analysis, TOAST prompts the user to another selection box where the main functionalities are presented again, and the bot is ready for another command.

The final score for each contributor is computed using a weighted system based on the values in the score column of Table 4. This system incorporates a scaling factor determined by the managers' responses to each survey question. Specifically:

- For a *strongly agree* response, a factor of 1 is applied, allocating the entire score to the relative community smell.
- An *agree* response results in a factor of 0.5, allocating half of the potential score.
- *Disagree* and *strongly disagree* responses are weighted similarly, with factors of -0.5 and -1, respectively.

[#] https://discord.com

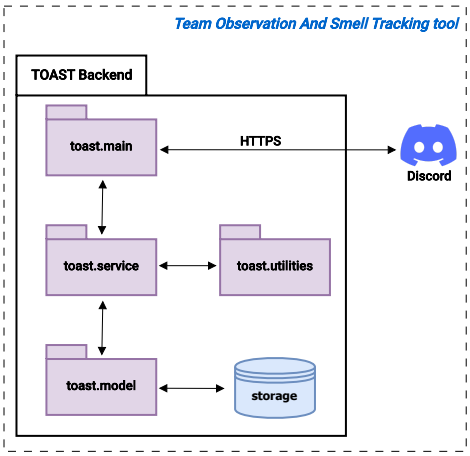

FIGURE 6 TOAST Architecture.

For example, if the user given the question *"The contributor has insufficient communication with the team"* (Table 4 ID 1) selects *disagree*, then a score of -1.856 is added to the final score, if they answer *strongly agree* the score added will be 3.712 and so on. The weighted scores for each question are then summed to produce the final score, standardized in a range of [0, 1] for each community smell. The score analysis serves as a quantitative measure of the contributor's susceptibility to various community smells. Moreover, the outcome of the analysis process produces final scores disaggregated by each analyzed, expressed in numerical form rather than as binary classifications. This approach eschews a simplistic susceptible/non-susceptible dichotomy for specific community smells. Using numerical scores empowered managers to establish context-specific thresholds for identifying contributors exhibiting particular smells. This flexibility is crucial given the heterogeneity of organizational contexts and situations, as establishing universally applicable thresholds presents a significant challenge within the constraints of currently available data. The numerical scoring system thus provides a flexible approach that accommodates the diverse environments in which community smells may manifest, allowing for nuanced interpretation and application of the results across varied team dynamics and organizational cultures.

5.2 | Tool Architecture and Technical Choices

As we already said, TOAST was developed as a Discord bot. Discord's widespread adoption within the software development community, where it has emerged as a robust alternative to enterprise communication tools like Slack and Microsoft Teams, has led to its selection as the primary interface. Discord's popularity among developers, rich API, and extensible bot ecosystem provides an ideal environment for deploying and scaling the tool.

For the development, we used `discord.js`,[‖] a JavaScript library that wraps the Discord API to streamline integration with JavaScript applications. This library benefits from robust open-source community support and offers extensive documentation and comprehensive guides that significantly facilitate development.

The architectural foundation of TOAST is built upon the Three-Tier pattern. This well-established software design paradigm stratifies the application into distinct functional layers, enhancing system maintainability and scalability. This architectural choice is particularly suited given the usage of the Discord platform. In this configuration, the Discord infrastructure manages the presentation layer, allowing the application to focus on core functionalities. Specifically, they are streamlined into three primary tasks: structuring message content, retrieving analytical results, and computing final scores. The module composition of the application is detailed in Figure 6.

[‖] https://discord.js.org/

5.3 | Empirical Evaluation of TOAST

To evaluate the usability and practical relevance of our tool, we conducted two complementary assessments of TOAST. First, a usability study was carried out to examine how easily users could navigate and interact with the system, ensuring that the interface supported efficient and intuitive use. Second, an experiment involving 11 participants with management experience was conducted to explore the tool's usefulness in real-world scenarios; specifically, its ability to support reflection, interpretation, and decision-making based on observed team behaviors.

5.3.1 | Usability Study

We applied *iterative usability testing*[35] to assess the overall usability of the graphical interface and the interaction patterns that were produced. This strategy is based on an iterative process in which, at each step, users give feedback about the tool's user interface—after executing a series of tasks—and developers modify the tool accordingly. We recruited five graduate students who attended and achieved a Human-Computer Interaction course during their degree. We asked them to conduct three tasks during each process iteration:

1. Start the TOAST bot in a Discord server and initiate the assessment process for a team member.
2. Add a new contributor to the team and then remove an existing contributor using the tool's team management features.
3. Answer the questions proposed for a team member using the provided interface to select responses.

After each iteration, we directly interviewed participants to measure the tool's usability using well-known instruments. Specifically, we adopted the System Usability Scale[36,37], the Questionnaire for User Interaction Satisfaction[38], and the NASA Task Load Index (NASA-TLX)[39]. We kept iterating the usability evaluation until reaching saturation[35].

Based on the feedback provided in the 4 iteration of the process, we improved the tool as follows:

- **Redesigned The Response Selection Mechanism:** The original dropdown menu for selecting responses was replaced with clickable buttons. This change aimed to streamline the response process and reduce the cognitive load on users.
- **Dynamic Question Display:** We implemented a feature to remove the question text once answered. This modification was designed to reduce visual clutter and help users focus on the current question at hand.
- **Color-Coded Response Buttons:** To enhance the visual distinction between different response options, we implemented a color-coding system. "Disagree" and "Strongly Disagree" buttons were colored red, while "Agree" and "Strongly Agree" buttons were colored green. This change aimed to provide visual cues about the nature of each response and prevent the perception that all options held equal weight.
- **Conversation Cleanup:** A feature was added to automatically delete the conversation containing the responses once the assessment was complete. This enhancement was intended to maintain confidentiality and reduce potential bias in future assessments.

Thanks to the feedback provided, we reached the final user interface, shown in Figure 1.

5.3.2 | Experiment on Practical Usefulness

To assess whether the proposed catalogue and tool can support everyday management tasks, we ran an *experiment*. The participants were recruited through *convenience sampling*[40], targeting project managers within our professional network.

We reached out via email and successfully enrolled 11 participants. From a demographic standpoint, five held a master's degree and six a doctoral degree in computer science. All were certified by the Project Management Institute (PMI) and had direct experience managing software development teams composed of 5 to 8 members. It is important to note that, although the study remains small in scale, involving certified project managers in empirical research poses inherent challenges, which naturally limits the size of the sample. That said, our objective was not to draw definitive conclusions. Rather, the goal was to explore the potential utility of the tool in realistic management scenarios and to contextualize the earlier survey and interview findings presented in the manuscript.

The experiment was conducted in person and consisted of three phases designed to simulate a realistic use case of the tool while allowing us to compare participants' unaided judgments with tool-supported evaluations. The overall goal was to understand whether TOAST could meaningfully support project managers in recognizing potential community smells in their teams, especially those that may not be immediately obvious. Rather than constructing an artificial or hypothetical scenario, we asked participants to reflect on the most recent software project they had personally managed. By grounding the evaluation in first-hand managerial experience, we ensured that participants' judgments were informed by authentic, context-rich interactions with their teams. It also allowed them to recall actual behaviors and dynamics, rather than reasoning about abstract or unfamiliar cases. This approach strengthened the ecological validity of the study and made the resulting comparisons between manual and tool-assisted assessments more meaningful.

The sessions were moderated by the first two authors of the paper, who facilitated the activities, answered any clarifying questions, and ensured that participants followed the experimental procedure correctly and independently. The remainder of the experiment followed a three-step sequence: (1) manual assessment, (2) tool-supported analysis, and (3) a follow-up semi-structured interview. Steps 1 and 2 were conducted in a single sitting, while Step 3 took place in a separate session scheduled after the tool outputs and manual responses were reviewed and analyzed. This separation allowed the moderators to tailor the interview questions to each participant's individual profile, with a focus on interpreting mismatches and reflecting on tool usefulness. In particular, the three steps were conducted as follows:

1. **Manual analysis.** After a brief introduction to the four community smells under investigation, each participant was provided with a spreadsheet pre-filled with the names (or pseudonyms) of the developers who had been part of the most recent project they had managed. The spreadsheet was organized to allow participants to assess each developer against the four community smells considered in the study. In this manual assessment phase, participants were instructed to rely solely on their personal recollections and managerial experience (i.e., without any support from the TOAST tool) to indicate whether, in their opinion, any team member had exhibited behaviors that aligned with each of the four smells. For each smell and team member, they were asked to respond with a binary "yes" or "no" judgment, accompanied by optional brief notes to explain or justify their decision. This unaided assessment served two purposes: (i) to capture the manager's baseline perception of social dynamics within the team, and (ii) to provide a reference point for later comparison with the tool-supported evaluation. The emphasis was on encouraging participants to reflect honestly and independently on their team members' attitudes, interactions, and working styles based on their managerial observations.
2. **Tool–supported analysis.** Next, the moderators introduced participants to the TOAST dashboard and demonstrated its basic functionalities in a brief walkthrough. This included how to create team member profiles, respond to the tool's symptom-based questions, and interpret the resulting smell scores. The goal was to ensure that each participant understood how the tool worked and could confidently complete the evaluation independently. Following the demonstration, participants were asked to input the same team members they had evaluated during the manual phase into the TOAST system. For each team member, the tool presented a series of targeted questions corresponding to symptoms drawn from the catalog associated with each of the four community smells. Participants answered these questions based on their recollection of the developer's behavior during the selected project. The tool then computed a smell score for each individual–smell pair, based on the weighted symptom responses.

   To ensure interpretability and consistency across all sessions, we adopted a straightforward thresholding rule: any smell score greater than or equal to 0.5 (on a continuous scale on $[-1; 1]$) was treated as a positive indication of that smell's presence. This threshold was chosen to correspond to the upper quartile of the possible range of values, reflecting a relatively strong signal of the smell. Smell scores below this threshold were considered either neutral or insufficiently indicative. All tool-generated smell detections were recorded for each team member and each smell, and later compared against the manager's initial, unaided judgments. This comparison enabled us to identify both convergences (true positives and true negatives) and divergences (false positives and false negatives) between the manager's perceptions and the tool's outputs. This comparison phase required post-hoc analysis and was therefore conducted separately from the interactive session, with step 3 (the interview) scheduled after this analysis had been completed.
3. **Semi–structured interview.** After completing both the manual and tool-assisted assessment tasks, each participant took part in a semi-structured interview conducted by one of the facilitators. These follow-up interviews were held individually and scheduled shortly after the experiment, once the comparison between the manual and tool-generated assessments had been completed. Each session lasted approximately 20–30 minutes and was structured around three key areas of inquiry:

- *Usability and intuitiveness of the tool.* Participants were asked to reflect on their experience interacting with the TOAST dashboard. They commented on the clarity of the interface, ease of navigation, responsiveness of the question-answering workflow, and overall learnability. Several participants also offered suggestions for improvement, such as customizable symptom thresholds or finer-grained explanations of the scoring mechanism.
- *Divergences between manual and tool-assisted assessments.* In this part, participants were shown the specific cases where their own smell assignments differed from those produced by the tool. They were asked to explain possible reasons for the discrepancies. This prompted insightful reflections on the role of personal bias, overlooked team dynamics, or contextual factors (e.g., organizational constraints) that may have shaped their initial judgments.
- *Perceived usefulness in real-world settings.* Finally, participants were invited to assess whether they would consider using TOAST in future projects and how they envisioned integrating it into their regular management workflows (e.g., during retrospectives or performance reviews).

From a quantitative perspective, we evaluated the alignment between participants' manual assessments and the tool-supported assessments produced by TOAST using standard information retrieval metrics. Specifically, we recorded 23 true positives, 4 false positives, 21 false negatives, and 37 true negatives. These results yield a precision of 0.88 and a recall of 0.53. In other words, in most cases where TOAST flagged a community smell, the participant had independently recognized a related issue: this suggests a strong degree of reliability in the tool's positive detections. However, the moderate recall indicates that TOAST missed several instances that managers had identified on their own, highlighting the continued necessity of human judgment in interpreting complex or context-specific team behaviors.

To complement these quantitative findings, we conducted a thematic analysis of the semi-structured interview transcripts. Several key insights emerged. In the first place, most participants agreed that TOAST offered credible recommendations. While not exhaustive in its detection capability, it was effective in highlighting overt or recurring patterns of social friction. Managers often treated the tool's output as a *"first warning system"*, i.e., a starting point for deeper reflection and investigation. With a precision of 0.85, participants felt confident in trusting the smells flagged by the tool as meaningful, even if not always complete. Secondly, several false negatives (i.e., smells identified by participants but not flagged by TOAST) were attributed to contextual subtleties that fall outside the scope of the tool's current symptom catalog. For example, one manager noted that certain interpersonal dynamics rooted in past team conflicts or time pressure during critical sprints were not captured by any specific symptom question. This limitation, while expected, emphasizes the need for possible future enhancements, such as more configurable or context-aware symptom tracking. Despite these limitations, the tool clearly succeeded in heightening managerial awareness. Nine out of eleven participants reported that using TOAST prompted them to reconsider their initial assessments and reflect on patterns they had previously overlooked. In some cases, participants admitted that they had underestimated persistent communication issues or behaviors that, in retrospect, warranted earlier intervention. This reflective aspect was particularly evident in the follow-up interviews, where several managers described how the tool encouraged them to adopt a more systematic approach to monitoring team interactions. Finally, three participants explicitly stated that they would have taken different mitigation actions had the tool been available during their most recent project. They indicated that the visibility of structured, symptom-based assessments would have helped them intervene earlier or frame discussions more constructively. Beyond immediate usage, others emphasized TOAST's potential as a decision-support aid, particularly for junior managers or teams with less experience in identifying social dysfunctions. Some also noted the benefit of having a persistent, structured record of team dynamics that could be revisited during retrospectives or performance evaluations.

Overall, the results of our experiment suggest that TOAST is best suited as a complementary tool in managerial decision-making. Rather than replacing the contextual insight of experienced managers, it provides a structured and reliable perspective that helps surface latent or previously overlooked social dynamics. Its high precision lends credibility to its alerts, while its limited recall underscores the continued importance of contextual awareness. Taken together, these qualities make TOAST a promising instrument—not only for supporting experienced managers in reflecting on team dynamics, but also for helping less experienced managers learn to identify teamwork problems in software projects.

6 | DISCUSSION, IMPLICATIONS, AND LIMITATIONS OF THE STUDY

In the following sections, we will discuss our research findings, their impact, and, ultimately, the threats to validity.

6.1 | Discussion

The analysis performed on the sample exposed in Section 4 revealed that, despite the significance of community smells in effective team management, 80.86% of the whole sample and 78.95% of the subset of managers were not familiar with them. We also found that, among the various characteristics associated with community smells, managers primarily perceived them as manifestations of communication breakdowns, but also highlighted a lack of transparency and reluctance to share knowledge. These patterns align with long-standing concerns in the management and organizational behavior literature, further suggesting that the concept of community smells, while unfamiliar in name, is grounded in issues practitioners already encounter in their daily work. The low recognition rate points to the fact that the technology transfer of community smell research into industrial practice is still in its infancy. As a result, structured awareness and proactive management of such phenomena remain difficult, which in turn increases the risk of deteriorating team dynamics and costly project failures.

This insight reinforces the rationale for our symptom-based approach: by focusing on concrete, observable behaviors rather than abstract taxonomies, we aim to bridge the gap between research and practice. This need was further validated through the analyses conducted in our study. Despite initial unfamiliarity with the terminology, project managers in the study were able to understand and reflect meaningfully on the behavioral symptoms presented via our tool. The structured output not only promoted critical assessment of their current team dynamics, but in some cases led to a reassessment of prior assumptions. These findings demonstrate that offering managers an interpretable lens through which to identify and reason about socio-technical issues can enhance their ability to recognize and respond to early warning signs.

> **Key Finding 1** — *The majority of managers interviewed were not familiar with the concept of community smells. However, they interpreted them primarily as communication problems, but also associated them with a lack of transparency and reluctance to share knowledge—indicating overlap with established management concerns.*

The outcomes of **RQ_1** illustrate that the current literature does not completely represent all behaviors contributors exhibit in the presence of community smells within software projects. This gap emerges from our finding that some meaningful symptoms surfaced only through practitioner feedback rather than prior studies. Specifically, three out of the ten validated symptoms were derived from managers' qualitative input rather than existing academic sources, underscoring the added value of field insights in refining smell detection. For instance, the behavior *"The contributor hoards critical knowledge and does not share it"* was included based on managers' responses, highlighting a frequent issue in team settings that is underrepresented in the literature and difficult to detect via automated tools.

Another observation is that managers often described symptoms that conceptually overlapped with one another or cut across different smell types. This reinforces the notion that social behaviors in teams rarely follow strict taxonomic boundaries. For example, reluctance to collaborate may indicate either *Lone Wolf* or *Prima Donna*, depending on context and intent. Rather than treating this overlap as a flaw, we interpret it as a strength of our approach: by cataloging concrete, observable behaviors rather than abstract categories, our method captures the multifaceted nature of real-world team dynamics. This flexibility is crucial for project managers, who must interpret behavioral patterns within the specific context of their teams rather than applying rigid definitions. The experiment results further support this view: even when participants were not familiar with the terminology of community smells, they were able to understand and act on the behavioral symptoms. This suggests that symptom-based representations may be more intuitively actionable than formal taxonomies, and they can help bridge the gap between empirical research and everyday managerial practice.

> **Key Finding 2** — *In response to **RQ_1**, we developed a refined catalog of concrete symptoms associated with key community smells (Table 4). Notably, some of the symptoms identified emerged directly from practitioner input rather than prior literature, revealing gaps in existing models and highlighting the value of managerial insights in surfacing overlooked yet practically relevant behaviors.*

The findings of **RQ_2** shed light on how project managers perceive the relevance of each identified symptom as an indicator of community smells. By analyzing quantitative responses across multiple iterations of the survey, we computed aggregated metrics—specifically, median values and standard deviations—for each symptom. These metrics serve a dual purpose: (1)

they quantify the extent to which each behavior is seen as indicative of a specific community smell, and (2) they make the catalog actionable, providing a prioritization signal that can guide practical interventions. As shown in Table 4, symptoms associated with *Lonesome Architecting* received the lowest relevance scores. This could be due to these behaviors being less frequently encountered or less easily observable in day-to-day project work. Importantly, the symptoms that emerged from practitioner feedback during the survey were rated as highly relevant—often on par with or exceeding those drawn from the literature—thereby reinforcing the value of incorporating real-world managerial insights into the catalog.

> **Key Finding 3** — *In response to **RQ$_2$**, we evaluated the relevance of each identified symptom by calculating metrics based on manager feedback across multiple survey iterations. The results revealed how strongly each symptom is perceived as indicative of community smells, with the analysis confirming that symptoms identified through managerial insights are just as significant as those documented in the literature.*

6.2 | Implications of the Study

While existing research on community smells has focused on detecting patterns at the team level, our study introduces a complementary perspective by identifying concrete, observable symptoms associated with four specific community smells. This symptom-based approach highlights issues that may originate from the behaviors or characteristics of individual team members, thereby increasing awareness and interpretability for practitioners. By making these socio-technical dynamics more tangible and accessible, our work offers important implications for both research and practice.

Implications for Practitioners. Our research insights have been applied to develop `TOAST`, a proof-of-concept recommendation system that makes our findings accessible to practitioners, thus enhancing technology transfer. Both the symptom catalog and `TOAST` have been developed to be complementary to automated analysis tools as described in Figure 1, since we envision that a combined approach can offer a more nuanced picture of the health of a development community—reducing the risk of hidden social debt and project delays. Unlike tools that produce binary classifications from large-scale repository mining, our approach helps project managers reason about concrete, individual-level behaviors that are often more immediately interpretable and actionable. The experiment we conducted provided insights into the practicality of our findings, indicating that our approach can enhance managerial awareness and decision-making even without prior exposure to the underlying terminology. Managers may also use our symptom catalog as a diagnostic checklist during performance reviews, team meetings, or project retrospectives, without necessarily relying on a tool like `TOAST`, making the catalog itself a lightweight and standalone resource. It is important to remember that *prevention is better than cure*, and prevention starts with education and awareness. According to Lehman's *"Conservation of Familiarity"* law[4], educating team members about the importance of healthy communication and collaboration is essential for the long-term success of software projects.

Implications for Researchers. We argue that *more behavioral literature needs to be developed in the context of community smells*, and our study provides a foundation for further exploration into this topic. Despite the growing presence of automated detection tools, some smells cannot be fully recognized without considering the interpersonal dynamics and contextual factors that shape team behavior. Our findings reinforce the importance of surfacing concrete, observable actions, rather than relying solely on abstract classifications, as a way to better align research outputs with real-world managerial concerns.

The symptom catalog we developed offers new avenues for expanding the scope of community smell detection beyond the current literature. Future research could focus on identifying additional symptoms across different project domains or team configurations, or on improving the methods used to assess symptom salience and impact. The iterative, practitioner-informed process used in this study may also inspire new methodological strategies that blend qualitative and quantitative inquiry to refine social-technical constructs.

Moreover, the open-source nature of our tool, `TOAST`, provides researchers with a practical platform for testing and validating new theories or models in diverse contexts. This tool can also serve as a vehicle for future user studies or observational experiments, potentially supporting interventions aimed at reducing social debt or improving collaboration. The ability to replicate and extend our study using the materials provided in our online appendix further contributes to the development of more sophisticated and context-aware community smell detection tools, advancing the field of software engineering and offering richer, more actionable insights for both practitioners and scholars.

6.3 | Limitations and Threats to Validity

This section reports the limitations and threats of the study, as well as the strategies adopted to address them. From a qualitative perspective, we mainly relied on validity and reliability criteria commonly discussed in qualitative research[41,42], while also considering threats specific to software engineering research[43].

Threats to Validity of the Research. As we conducted a mixed-method study involving both survey-based data and qualitative analysis, we faced potential threats such as researcher bias and respondent bias[41]. Regarding *researcher bias*, we adopted triangulation and peer debriefing. Our survey instrument was grounded in, and iteratively compared against, the existing peer-reviewed literature on community smells. This continuous alignment helped prevent conceptual drift. Moreover, frequent peer debriefings between the first two authors ensured that subjective interpretations were discussed and critically assessed within the team. *Respondent bias* was mitigated using multiple strategies. First, we designed the study with attention checks and mixed question formats (both open- and close-ended) to reduce the risk of inauthentic responses or the use of external tools like generative AI. Second, we employed a multi-stage process (prolonged involvement) involving multiple iterations and diverse participant pools to validate the robustness and consistency of the findings.

For the experiment, internal validity may be affected by participants' interpretation of symptoms or their *social desirability bias*, i.e., the tendency to report team dynamics in a favorable light. To mitigate this potential limitation, we framed the task in reflective terms, encouraging critical thinking rather than evaluative judgment. Additionally, it is worth remarking that the experiment was grounded in participants' real-world experience: each manager was asked to reason about the dynamics of the most recent team they had managed. This context-rich approach increased the ecological validity of the study and allowed participants to anchor their reflections in concrete, familiar settings. While this limits experimental control, it significantly enhances the authenticity and depth of the insights gathered.

Threats to Reliability of the Research. Reliability refers to the extent to which the study processes and outcomes are consistent and replicable[42]. We addressed this by providing a detailed description of our methodology, including participant demographics, recruitment strategies, and the iterative design process. We also supported our findings with both quantitative summaries and qualitative quotes from participants' open-ended responses. Importantly, we made all instruments, raw data, and analyses available in a public online appendix[23], allowing other researchers to replicate or extend our work.

For the experiment, reliability may be influenced by the relatively small sample size (11 participants). However, recruiting managers for empirical research is inherently challenging, and we view the successful involvement of professionals as a strength of this study. All participants had managerial experience and reflected on real teams they had recently managed, enhancing the authenticity of the results. While future replications with larger or more diverse samples could further reinforce generalizability, our experiment offers a concrete and replicable setup for evaluating how practitioners interpret and respond to smell-related behavioral symptoms.

Threats to Transferability of the Research. Transferability refers to the extent to which findings can be applied to other contexts[41]. To support transferability, we selected participants with varied professional roles and geographic locations using the PROLIFIC platform. We also ensured representation of both general practitioners and managers, tailoring each research question to the appropriate subset (e.g., $\mathbf{RQ_1}$ and $\mathbf{RQ_2}$ to managers).The experiment involved small-team managers reflecting on their own recent projects, allowing for grounded, experience-based insights. While this provided strong ecological validity, future work could explore whether the findings generalize to larger-scale initiatives, cross-functional teams, or remote development settings, where team dynamics may differ substantially.

7 | CONCLUSIONS AND FUTURE WORK

Our study introduces a complementary approach to detecting community smells, focusing on symptom recognition at the contributor level rather than the more common team-level interactions. We employed a mixed-method approach, combining a literature review with a practitioner survey, to identify and validate a catalog of symptoms associated with four specific community smells. The catalog provides managers with a practical resource for early intervention, complementing existing detection methods and enhancing overall community health management. To support this approach, we developed `TOAST`, a tool designed to assist managers in monitoring and addressing these issues. We then assessed the usability of the tool using an iterative usability testing and performed a experiment to assess the practical usefulness of the tool in real management scenarios.

The participants to the study reported that the tool have raised their awareness on team members behavior that they would have resolved differently.

Future research will focus on expanding the current symptom catalog to cover additional contributor-level community smells that were not included in this study. Building on the foundation established here, we plan to conduct a more systematic investigation into the remaining smells identified in the literature, particularly those that, while not explicitly framed as individual-level in prior work, may nonetheless manifest through observable individual behaviors recognizable by a project manager. For instance, several additional smells, including *Cognitive Distance*, *Newbie Free-Riding*, *Leftover Techie*, *Organizational Silo*, and *Radio Silence*, could plausibly be inferred through a manager's awareness of communication patterns, interpersonal tensions, or team feedback. These insights reinforce our goal of expanding the catalog with additional symptoms that reflect real-world team dynamics and social frictions. Extending the symptom set in this way will also open the door to developing refined detection metrics informed by practitioner feedback and empirical observations. Where appropriate, these advancements may be integrated into existing detection pipelines (e.g., CSDETECTOR [20]) or incorporated into new functionalities within TOAST, such as context-aware sensitivity tuning or cross-platform integration (e.g., with communication tools like Teams or Slack). In the long term, we aim to evolve TOAST into a configurable decision-support system that project managers can adapt to the specific dynamics of their teams. This includes incorporating historical views, trend dashboards, and symptom-based alerts that can help identify emergent social issues early. Finally, we also see value in exploring how the symptom catalog might support proactive management practices, such as structured retrospectives or targeted interventions, to prevent the emergence of community smells and improve collaboration within software teams.

References

1. Institute PM. *A Guide to the Project Management Body of Knowledge*. 7 ed., 2021.
2. Brooks Jr FP. *The mythical man-month: essays on software engineering*. Pearson Education, 1995.
3. Ralph P, Chiasson M, Kelley H. Social Theory for Software Engineering Research. In: EASE '16. Association for Computing Machinery 2016; New York, NY, USA
4. Lehman MM. On understanding laws, evolution, and conservation in the large-program life cycle. *Journal of Systems and Software.* 1979;1:213–221.
5. Gupta A, Sharma S. Software maintenance: Challenges and issues. *Issues.* 2015;1(1):23–25.
6. Ulziit B, Warraich ZA, Gencel C, Petersen K. A conceptual framework of challenges and solutions for managing global software maintenance. *Journal of Software: Evolution and Process.* 2015;27(10):763–792.
7. DeMarco T. *Controlling software projects: Management, measurement, and estimates*. Prentice Hall PTR, 1986.
8. Tamburri DA, Kruchten P, Lago P, Vliet Hv. What is social debt in software engineering?. *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE).* 2013:93-96.
9. Fowler M. *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 2018.
10. Caballero-Espinosa E, Carver C. J, Stowers K. Community smells—The sources of social debt: A systematic literature review. In:
11. 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.* 2021;47(1):108-129. doi: 10.1109/TSE.2018.2883603
12. Martini A, Bosch J. Revealing social debt with the CAFFEA framework: An antidote to architectural debt. In: IEEE. 2017:179–181.
13. Tamburri DA, Palomba F, Serebrenik A, Zaidman A. Discovering community patterns in open-source: a systematic approach and its evaluation. *Empirical Software Engineering.* 2019;24(3):1369–1417.
14. Tamburri DA, Palomba F, Kazman R. Exploring Community Smells in Open-Source: An Automated Approach. *IEEE Transactions on Software Engineering.* 2019;47(3):630-652. doi: 10.1109/TSE.2019.2901490
15. Catolino G, Palomba F, Tamburri DA, Serebrenik A, Ferrucci F. Gender Diversity and Women in Software Teams: How Do They Affect Community Smells?. *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society.* 2019.
16. Lambiase S, Catolino G, Tamburri DA, Serebrenik A, Palomba F, Ferrucci F. Good Fences Make Good Neighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells. 2022.
17. Palomba F, Tamburri DA. Predicting the emergence of community smells using socio-technical metrics: a machine-learning approach. *Journal of Systems and Software.* 2021;171:110847.

18. Almarimi N, Ouni A, Chouchen M, Saidani I, Mkaouer MW. On the detection of community smells using genetic programming-based ensemble classifier chain. In: 2020:43–54.

19. Voria G, Pentangelo V, Della Porta A, et al. Community Smell Detection and Refactoring in SLACK: The CADOCS Project. *Information and Software Technology.* 2022;146:106853.

20. Almarimi N, Ouni A, Chouchen M, Mkaouer MW. csDetector: an open source tool for community smells detection. *ESEC/FSE 2021.* 2021. doi: 10.1145/3468264.3473121

21. Banaeianjahromi N, Smolander K. Lack of communication and collaboration in enterprise architecture development. *Information Systems Frontiers.* 2019;21(4):877–908.

22. Lathrop D, Ruma L. *Open government: Collaboration, transparency, and participation in practice*. " O'Reilly Media, Inc.", 2010.

23. Della Porta A, Lambiase S, Catolino G, Ferrucci F, Palomba F. Online Appendix - A Novel, Tool-Supported Catalog of Community Smell Symptoms. https://github.com/atdepo/toast-tool; .

24. Tamburri DA, Kruchten P, Lago P, Vliet H. Social Debt in Software Engineering: Insights from Industry. *Journal Loof Internet Services and Applications.* 2015. doi: 10.1186/s13174-015-0024-6.

25. Reid B, Wagner M, d'Amorim M, Treude C. Software Engineering User Study Recruitment on Prolific: An Experience Report. *arXiv preprint arXiv:2201.05348.* 2022.

26. Kitchenham BA, Pfleeger SL. Personal Opinion Surveys. In: , , Springer, 2008:63–92.

27. Andrews D, Nonnecke B, Preece J. Conducting research on the internet:: Online survey design, development and implementation guidelines. 2007.

28. Flanigan TS, McFarlane E, Cook S. Conducting survey research among physicians and other medical professionals: a review of current literature. In: . 1. 2008:4136–47.

29. Tamburri DA. Software Architecture Social Debt: Managing the Incommunicability Factor. *IEEE Transactions on Computational Social Systems.* 2019;6.

30. Annunziata G, Ferrara C, Lambiase S, et al. An Empirical Study on the Relation Between Programming Languages and the Emergence of Community Smells. In: IEEE. 2024:268–275.

31. Wicks D. The coding manual for qualitative researchers. *Qualitative research in organizations and management: an international journal.* 2017;12(2):169–170.

32. Salas E, Shuffler ML, Thayer AL, Bedwell WL, Lazzara EH. Understanding and improving teamwork in organizations: A scientifically based practical guide. *Human resource management.* 2015;54(4):599–622.

33. Bell ST, Brown SG, Colaneri A, Outland N. Team composition and the ABCs of teamwork.. *American psychologist.* 2018;73(4):349.

34. Palomba F, Tamburri DA. Predicting the emergence of community smells using socio-technical metrics: a machine-learning approach. *Journal of Systems and Software.* 2021.

35. Genov A. Iterative usability testing as continuous feedback: A control systems perspective. *Journal of Usability Studies.* 2005;1(1):18–27.

36. Klug B. An overview of the system usability scale in library website and system usability testing. *Weave: Journal of Library User Experience.* 2017;1(6).

37. Grier RA, Bangor A, Kortum P, Peres SC. The system usability scale: Beyond standard usability testing. In: . 57. SAGE Publications Sage CA: Los Angeles, CA. 2013:187–191.

38. Maryland HCILUO. "QUIS: The Questionnaire for User Interaction Satisfaction.". .

39. Hart SG, Staveland LE. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In: , , . 52. Elsevier, 1988:139–183.

40. Golzar J, Noor S, Tajik O. Convenience sampling. *International Journal of Education & Language Studies.* 2022;1(2):72–77.

41. Lincoln YS, Guba EG. *Naturalistic inquiry*. sage, 1985.

42. Robson C. Real world research: A resource for social scientists and practitioner-researchers. *Oxford, UK: Blackwell Publishers.*. 2002.

43. Seaman CB. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering.* 1999;25(4):557–572.