



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**ANDRÉ FILIPE QUEIROZ DE MELO E SOARES**

**COMMUNITY SMELLS IN SOFTWARE TEAMS: A SYSTEMATIC  
LITERATURE MAPPING**

**CAMPINA GRANDE - PB**

**2023**

**ANDRÉ FILIPE QUEIROZ DE MELO E SOARES**

**COMMUNITY SMELLS IN SOFTWARE TEAMS: A SYSTEMATIC  
LITERATURE MAPPING**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**Orientador(a): Professor Tiago Lima Massoni**

**CAMPINA GRANDE - PB**

**2023**

**ANDRÉ FILIPE QUEIROZ DE MELO E SOARES**

**COMMUNITY SMELLS IN SOFTWARE TEAMS: A SYSTEMATIC  
LITERATURE MAPPING**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr.(a.) Tiago Lima Massoni  
Orientador e Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Professora Dr.(a.) Francisco Vilar Brasileiro  
Examinador – UASC/CEEI/UFCG**

**Professora Dr.(a.) Marcelo Alves de Barros  
Examinador – UASC/CEEI/UFCG**

**Trabalho aprovado em: 14 de Fevereiro de 2023.**

**CAMPINA GRANDE - PB**

## **ABSTRACT**

Community smells are symptoms of organizational and social problems in software teams that often increase project costs and affect software quality. Recent research has identified several community smells and defined them as suboptimal patterns related to organizational social structure in software development teams, such as lack of communication, coordination, and collaboration. For this reason, this study aims to conduct a systematic mapping from scientific articles in order to understand how the topic is being treated. In view of this, 24 articles were selected that address the subject in several existing databases and classification criteria were elaborated with the objective of answering the questions made and evaluating the results, this article concluded that studying community smells at the structural and organizational level can be vital to freeing up software teams to avoid critical organizational failures that may in the future entail considerable additional costs.

**KEYWORDS:** Systematic Mapping, Community Smells, Software Development.

# COMMUNITY SMELLS IN SOFTWARE TEAMS: A SYSTEMATIC LITERATURE MAPPING

André Filipe Queiroz de Melo e Soares

Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil

andre.soares@ccc.ufcg.edu.br

Tiago Lima Massoni

Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil

massoni@dsc.ufcg.edu.br

## RESUMO

*Community smells* são sintomas de problemas organizacionais e sociais em equipes de software que frequentemente aumentam os custos do projeto e afetam a qualidade do software. Pesquisas recentes identificaram vários *community smells* e os definiram como padrões abaixo do ideal relacionados à estrutura social organizacional em equipes de desenvolvimento de software, como falta de comunicação, coordenação e colaboração. Por esse motivo, esse estudo tem como objetivo realizar um mapeamento sistemático a partir de artigos científicos com a finalidade de compreender como o tema está sendo tratado. Em vista disso, foram selecionados 24 artigos que abordam a temática em diversas bases de dados existentes e elaborado critérios de classificação com o objetivo de responder às indagações feitas e avaliando os resultados, este artigo concluiu que estudar os *community smells* no nível estrutural e organizacional pode ser vital para livrar as equipes de software para evitar falhas organizacionais críticas que podem no futuro acarretar em custos adicionais consideráveis.

## PALAVRAS-CHAVE

Mapeamento Sistemático, Community Smells, Desenvolvimento de Software.

## 1. INTRODUÇÃO

Atualmente, a economia é fortemente dependente dos softwares. Tem sido cada vez mais comum empresas que não comercializam software desenvolverem elas para suportarem seus negócios[1]. De modo geral, as

empresas caminham se desenvolvendo, enfrentando o tempo para que assim permaneçam relevantes no mercado.

A engenharia de software depende cada vez mais do bem-estar de grandes comunidades distribuídas globalmente e de suas redes sociais no desenvolvimento de software. Um maior entendimento da estrutura organizacional dessas comunidades e suas características sociais, e os fatores que afetam sua qualidade, são fundamentais para o sucesso dos projetos de software[2].

Ao considerarmos as interações entre as pessoas num ambiente de trabalho tem de se levar em conta que as pessoas não são previsíveis e que, muitas vezes, o comportamento do outro nem sempre será aquele que se espera dentro da coletividade, influenciando assim o produto final. De maneira que, tanto a eficiência da equipe quanto a qualidade organizacional estão relacionadas com a comunicação.

Nessa perspectiva, surgem os *community smells*, os quais são definidos como um conjunto de circunstâncias que ocorrem dentro da comunidade de desenvolvimento de software, tendo um efeito negativo sobre as relações de dentro das equipes[5], além do mais, eles estão ligados a circunstâncias baseadas em má organização e práticas sociais[5].

Esses problemas podem se manifestar de várias formas, por exemplo, falta de comunicação, colaboração ou coordenação entre os membros de uma comunidade de desenvolvimento de software.

Por exemplo, um dos *community smells* comuns da comunidade é o “efeito de silo organizacional” [13] que se manifesta como uma subestrutura de rede social recorrente apresentando uma estrutura de comunidade de desenvolvedores altamente desacoplada.

Em vista disso, o mapeamento sistemático da literatura é comumente utilizado como forma de buscar, avaliar e interpretar as pesquisas disponíveis, as quais são relevantes para o tema. Este tipo de estudo possui um foco na categorização do tópico de pesquisa, levando em conta também, a condução de mapeamentos sistemáticos em tópicos de pesquisa nos quais poucas evidências estão disponíveis na literatura[8].

Com o objetivo de buscar diminuir esta lacuna, neste trabalho apresentamos um estudo de mapeamento sistemático sobre *community smells*. O objetivo é unificar conhecimentos dispersos sobre o tema a partir de fontes literárias a fim de analisar resultados de pesquisas na área, destacando o que está consolidado na literatura e refletir sobre as limitações do conhecimento existente.

Para atingir esses objetivos, formulamos as seguintes perguntas de pesquisa, com critérios de seleção para a análise de cada artigo, a fim de verificar se eles são relevantes o suficiente para ajudar a responder às questões de pesquisa definidas:

- **Q1: Quais ferramentas de detecção existentes na literatura permitem identificar a presença de *community smells* ?**
- **Q2: Quais são os *community smells* que ocorrem com mais frequência nas equipes de software?**

Na próxima seção, apresentaremos os conceitos necessários para uma maior compreensão sobre a problemática, as quais foram utilizados para a nossa pesquisa. Em seguida, detalhamos a metodologia do estudo, como foi realizada a nossa análise.

O restante deste artigo está organizado da seguinte forma: na seção 2, fornecemos uma explicação teórica

sobre os assuntos abordados neste trabalho e os conceitos utilizados em nosso estudo para ajudar na compreensão das seções seguintes. Na seção 3, é apresentada a metodologia utilizada neste estudo à medida que coletamos os trabalhos e os analisamos. Na seção 4, são discutidos os resultados obtidos e a análise realizada sobre os dados coletados. Por fim, na seção 5, são apresentadas as conclusões e considerações finais.

## 2. FUNDAMENTAÇÃO TEÓRICA

A engenharia de software é inerentemente uma atividade social, muitas vezes envolvendo organizações, gerentes, desenvolvedores e partes interessadas de todo o mundo para compartilhar ações coletivas. Isso leva a vários desafios para os desenvolvedores, especialmente quando se trata de colaboração e comunicação, como conflitos entre indivíduos, problemas com o idioma de comunicação e barreiras culturais.

Sabendo disso e atrelado ao fato que um produto de software é resultado de uma trabalho harmônico entre a equipe, contudo *community smells* podem ocorrer, ou seja, condições em que há um subaproveitamento da organização e comunicação da(s) equipe(s) podendo levar a custos imprevisto do projeto[5].

A estrutura social organizacional em uma comunidade de desenvolvimento de software, incluindo as interações entre os desenvolvedores, é um pré-requisito essencial para um produto de software bem-sucedido. Aliás, quando um software apresenta um problema em seu código, pode ser definido como sintomas de más escolhas de projeto e implementação[4].

Problemas relacionados à comunidade são realmente percebidos como indicadores da persistência de *smells* no código, indicando assim a existência de outros aspectos que impactam a manutenção de falhas de implementação técnica[6]. Estudos recentes cunham o conjunto de padrões sociotécnicos que impactam negativamente na saúde organizacional do projeto, como *community smells* [6].

Tamburri[5] define *community smells* como “conjuntos de circunstâncias organizacionais e sociais, tendo relações causais implícitas”, os quais produzem principalmente emoções fortes, estressam as interações sociais, afetam o desempenho da equipe e diminuem a qualidade do software. Assim, enquanto permanecerem no local, os *community smells* resultam em custos adicionais para o projeto, os custos em si de uma má comunicação, por exemplo, são estimados em U\$S 37 bilhões[11]. Alguns exemplos existentes no estado da arte são:

- Silo organizacional: Grupos isolados na comunidade que não se comunicam entre si, exceto por meio de um ou dois de seus respectivos membros[5];
- Nuvem escura: Uma sobrecarga excessiva de informações devido à falta de comunicação estruturada, o que pode levar a um grande aumento de trocas de dados em uma comunidade [5];
- Bottleneck(Gargalo): Um dos membros se interpõe em todas as interações formais em duas ou mais subcomunidades com pouca ou nenhuma flexibilidade para introduzir outros canais paralelos [5];
- Time-warp: Colegas experientes assumem erroneamente prazos para troca de comunicações e nenhuma necessidade de coordenação explícita[9];
- Distância Cognitiva: Distância que os desenvolvedores percebem nos níveis físico, técnico, social e cultural em relação a pares com diferenças consideráveis de formação[9];
- DevOps Clash: Confrontos entre equipes de desenvolvimento e operações dentre várias localizações geográficas com obrigações contratuais para atividades de desenvolvimento ou operações[9];
- Lobo Solitário: Indivíduos inflexíveis que realizam seu trabalho independentemente ou sem considerar seus colegas[14];

- Missing Link: Quando os desenvolvedores trabalham isoladamente sem se comunicarem com seus pares[5][3].

Optou-se por iniciar introduzindo por esses tipos, pois eles serão novamente mencionados no decorrer deste trabalho.

Nos últimos anos, a pesquisa sobre o *community smells* gradualmente ganhou atenção. Trabalhos recentes estabeleceram quais são os principais fatores que afetam os *community smells*[15][7] e como os desenvolvedores podem refatorar os *community smells* na prática[14]. Como por exemplo, Tamburri [13] mostrou que *community smells* são altamente difusos e considerados nocivos, e que alguns fatores sociotécnicos existentes estão associados à sua presença.

### 3. METODOLOGIA

Durante a realização de um mapeamento da literatura é imprescindível obedecer a certos passos, que conforme Kitchenham e Charters[8] segue os mesmos padrões da análise sistemática da literatura, no entanto, sem a investigação minuciosa e detalhada exigida para a mesma. Assim, as etapas são: (1) determinar a questão de pesquisa; (2) coleta de materiais nas bases de dados bibliográficos; (3) elaboração de estratégias para busca avançada; (4) seleção dos textos e sistematizar as informações encontradas.

Vale ressaltar que as fases do processo de condução são iterativas, ou seja, as atividades são refinadas conforme o andamento do processo, por exemplo, a definição dos critérios de seleção começa durante o planejamento, mas novos critérios podem ser identificados durante as atividades de seleção[8].

#### 3.1 Planejamento e questões de pesquisa

Como parte do mapeamento do sistema de literatura, foi criado um protocolo de mapeamento para o mesmo ao iniciar a fase de planejamento realizada neste trabalho. Propomos questões de pesquisa, critérios de seleção e outras especificações para uma conduta bem-sucedida.

Entre outros métodos de pesquisa, as questões de pesquisa de um mapeamento da literatura norteará a condução de todo o estudo. Portanto, estruturamos nosso estudo em torno de duas questões principais de pesquisa em que todas as atividades subsequentes decorrem da questão de pesquisa, por isso é crucial defini-las fielmente com o tópico.

Para detectar com antecedência instâncias potenciais de má gestão da comunidade durante um ciclo de vida de desenvolvimento de software, técnicas eficientes e automatizadas são necessárias [5,18]. Em vista disso, surge nosso primeiro questionamento, buscando entender quais opções já elaboradas e analisadas existem na literatura, a fim de fornecer alternativas de mitigar os efeitos colaterais de um *community smell*.

- **Q1: Quais ferramentas de detecção existentes na literatura permitem identificar a presença de *community smells* ?**

Os pesquisadores têm estudado os padrões e *community smells* isoladamente, por exemplo, avaliando como os *community smells* se manifestam e podem ser mitigados[7,12]. Nesse entendimento, a nossa segunda pergunta investiga a relação entre os *community smells* e sua ocorrência na prática, para entender se certos *community smells* tendem a ocorrer com mais frequência do que outros.

- **Q2: Quais são os *community smells* que ocorrem com mais frequência nas equipes de software?**

Além do mais, os artigos coletados foram catalogados no programa Zotero<sup>1</sup>, que permite a classificação automática de várias informações relativas à pesquisa, para gerenciar dados bibliográficos e materiais relacionados à temática, facilitando a quantificação de pesquisas.

### 3.2 Coleta de materiais

Os estudos encontrados foram preferencialmente escolhidos no idioma inglês devido a sua abundância de conteúdo acerca do tema, no entanto não

desconsiderando caso haja artigos na língua portuguesa.

Dentre as mais diversas fontes de artigos científicos, foi escolhido as bibliotecas digitais, as quais é possível acessar livremente e gratuitamente, tais como: IEEE e ACM Digital Library. Por outro lado, com o Google Scholar é possível restringir e personalizar os resultados da pesquisa para encontrar exatamente o que deseja, utilizando caracteres como o de adição(+), subtração(-), aspas(“”) e entre outros.

Além disso, primeiro foi realizada uma busca manual para obter conhecimento sobre o assunto, seguida de uma busca automática, utilizando repositórios bibliográficos, com apoio computacional, para fornecer um conjunto de informações e os artigos retornados.

### 3.3 Elaboração das estratégias de busca

A primeira tarefa da pesquisa foi criar inicialmente palavras chave, tais como “community smells”, “software development”, “software teams”. Os repositórios de artigos das bibliotecas digitais explicadas no tópico 3.2, em específico a IEEE e ACM Digital Library, oferecem suporte à termos de pesquisa usando os conectores "AND" (AND lógico) e "OR" (OR lógico). Dessa forma, define-se uma string de busca, podendo ser vista na tabela 1, os termos a serem buscados, de forma que leva em consideração a questão de interesse da pesquisa, antes de iniciar a busca propriamente dita.

A propósito, com o Google Scholar o seu processo de busca ocorreu de maneira levemente diferente, com a utilização do símbolo de adição(+), o qual é usado para incluir e encontrar palavras correspondentes, fazendo com que se busque diversas palavras específicas que podem estar relacionadas, tornando a busca mais precisa. Vale lembrar que as palavras se encontram dentro de aspas, pois retornam exatamente o que está escrito.

É importante enfatizar que o processo de definição das strings de busca é iterativo, envolvendo múltiplos ciclos de experimentos, validação dos artigos retornados e ajuste das strings de busca [8].

---

<sup>1</sup> <https://www.zotero.org/>

Bases de busca	String de busca
Google Scholar	"community smells" + "software development" + "software teams"
ACM Digital Library	("community smells") AND ("software development") OR ("software teams")
IEEE Xplore	("community smells") AND ("software development") OR ("software teams")

Tabela 1: strings de buscas usados nas bibliotecas digitais

### 3.4 Seleção dos artigos

Os critérios de inclusão/exclusão precisam ser especificados, já que os *community smells* são um tópico relativamente novo quando se trata de desenvolvimento de software, escolhendo artigos de acordo com a sua relevância e quantidade de citações, evitando estudos duplicados, inacabados e não publicados. Além disso, é fundamental salientar para um critério especial a respeito do intervalo de tempo, o qual foi definido para a seleção dos trabalhos o intervalo entre 2013 até 2022, pois o artigo seminal sobre esse tema foi publicado em 2013[6].

Ao longo do processo de mapeamento, foram retornados 101 resultados no Google Scholar, 940 no ACM Digital Library e 281 no IEEE Xplore, os quais após uma triagem, 24 obras relevantes ao assunto foram selecionadas, todas em inglês.

Tais trabalhos foram categorizados como empíricos, os quais há evidências empíricas que apoiam diretamente os resultados da pesquisa ou como teóricos, em que a compreensão dos pesquisadores sobre um tópico ou campo apoia os resultados da pesquisa, obras as quais podem ser vistas na tabela 2.

Além do mais, todos os trabalhos foram lidos e analisados para entender os objetivos de cada trabalho, o contexto em que foram analisados e

desenvolvidos para verificar como cada estudo relacionou os temas com os métodos de pesquisa utilizados e os principais achados encontrados.

## 4. RESULTADOS

Em primeira análise, foram lidos os resumos(*abstracts*), palavras-chave, introdução e conclusões de cada estudo e ao aplicar os critérios de inclusão e exclusão para examinar os artigos, essa triagem identificou 24 estudos para serem estudados.

Tabela 2: Estudos coletados durante o mapeamento

#	Título	Ano	Tipo do Estudo
1	A preliminary study on the assignment of GitHub issues to issue commenters and the relationship with social smells	2022	Empirico [21]
2	An empirical study on the effect of community smells on bug prediction	2021	Empirico [22]
3	Analyzing the Relationship between Community and Design Smells in Open-Source Software Projects: An Empirical Study	2022	Empirico [10]
4	Beyond Technical Aspects: How Do Community Smells Influence the Intensity of Code Smells?	2018	Empirico [13]
5	Community Smell Occurrence Prediction on Multi-Granularity by Developer-Oriented Features and Process Metrics	2022	Empirico [23]
6	csDetector: an open source tool for community	2021	Empirico

	smells detection		[24]
7	Do Missing Link Community Smell Affect Developers Productivity: An Empirical Study	2021	Empirico [3]
8	Exploring Community Smells in Open-Source: An Automated Approach	2019	Empirico [19]
9	Gender Diversity and Community Smells: A Double-Replication Study on Brazilian Software Teams	2022	Empirico [25]
10	Gender Diversity and Community Smells: Insights From the Trenches	2019	Empirico [7]
11	Gender Diversity and Women in Software Teams: How Do They Affect Community Smells?	2019	Empirico [12]
12	Good Fences Make Good Neighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells	2022	Empirico [26]
13	How do community smells influence code smells?	2018	Empirico [27]
14	Impacts of software community patterns on process and product: An empirical study	2022	Empirico [17]
15	Learning to detect community smells in open source software projects	2020	Empirico [28]
16	On the detection of community smells using genetic	2020	Empirico

	programming-based ensemble classifier chain		[18]
17	Predicting Community Smells' Occurrence on Individual Developers by Sentiments	2021	Empirico [29]
18	Predicting the emergence of community smells using socio-technical metrics: A machine-learning approach	2021	Empirico [30]
19	Refactoring community smells in the wild: the practitioner's field manual	2020	Empirico [14]
20	Social debt in software engineering: insights from industry	2015	Empirico [5]
21	Software Architecture Social Debt: Managing the Incommunicability Factor	2019	Empirico [31]
22	Splicing Community and Software Architecture Smells in Agile Teams: An industrial Study	2019	Empirico [9]
23	Splicing Community Patterns and Smells: A Preliminary Study	2020	Empirico [15]
24	Understanding Community Smells Variability: A Statistical Approach	2021	Empirico [16]

Durante a pesquisa, foram formuladas uma série de questões de pesquisa que nortearam este trabalho. Para responder a essas questões, foram identificadas pesquisas online abordando o tema “*community smells*”.

Uma observação a ser levantada, sendo aplicada aos questionamentos feitos, é que diferentes padrões de

comunidade se relacionam com diferentes *community smells*. Com isso, as razões para a presença de *community smells* é a de que estão fortemente relacionadas às características e peculiaridades dos padrões da comunidade[15][17].

- **Q1: Quais ferramentas de detecção existentes na literatura permitem identificar a presença de *community smells* ?**

A literatura descreve ferramentas que fornecem um ponto de partida para automatizar e gerenciar de forma mais eficaz os *community smells*. Os recursos dessas ferramentas incluem identificar, antecipar e monitorar o impacto de tal problemática. Primeiro, a ferramenta implantada CodeFace4Smells identifica os *community smells* em projetos de código aberto, analisando o histórico de commits dos contribuidores para determinar os efeitos dos *community smells* na colaboração. Introduzido no artigo “Exploring Community Smells in Open-Source: An Automated Approach”, sua capacidade atual é limitada a quatro *community smells*: nuvens escuras, lobos solitários, silos organizacionais e silêncio de rádio [19]. O CodeFace4Smells demonstra ter alta precisão na detecção do impacto dos *community smells* na comunicação e colaboração em interações sociais de desenvolvedor.

A ferramenta também apoia o estudo dos *community smells* e sua relação com a força dos *smells* no código [13] e equipes com diversidade de gênero [12].

Além disso, outra ferramenta existente é a YOSHI, apresentada no estudo denominado “Discovering community patterns in open-source: a systematic approach and its evaluation”. Tal ferramenta extrai informações de repositórios de projetos de código aberto para identificar tipos de comunidades de engenharia de software e tipos de comunidade [20]. Para fins exemplificativos, YOSHI pode usar o controle de acesso ao repositório para estabelecer um nível de formalidade. Um alto grau de formalidade pode levar a *community smells*, como o silêncio do rádio(radio silence).

Ademais, temos na literatura o csDetector, um método de detecção baseado em aprendizado de máquina que analisa más práticas de desenvolvimento de software[48]. O csDetector detecta *community smells* com uma precisão média de 96% em 74 projetos de código aberto. O modelo se comparado a outros também supera o CodeFace4smell com uma alta precisão de detecção de 94% para 98%. Este método automatizado aprende e classifica oito dos nove *community smells* definidos por Tamburri[5].

Portanto, vemos essas ferramentas como desenvolvimentos avançados que verificam automaticamente os repositórios de projetos de software e identificam os *community smells*. Contudo, apesar de tais ferramentas serem um avanço, elas cobrem um conjunto limitado de *community smells*, sendo fundamental futuramente expandir seu escopo para serem mais abrangentes. Além disso, essas ferramentas requerem mais desenvolvimento, análise em vários contextos diferentes e avaliações de usabilidade.

- **Q2: Quais são os *community smells* que ocorrem com mais frequência nas equipes de software?**

Os *community smells* são fenômenos emergentes a serem considerados e abordados em conjunto com a qualidade dos artefatos de software sendo produzidos e mantidos por equipes de software[17]. Após análise dos artigos catalogados durante o mapeamento da literatura, percebeu-se que não há uma unanimidade acerca de qual *community smells* existente na literatura é o mais frequente nas equipes de software.

Variando de acordo com o contexto da situação e como é estruturado a organização, certos tipos de *community smells* podem ocorrer, por exemplo:

- Silo organizacional;
- Missing Links;
- Lobo Solitário;

Dessa maneira, observou-se que suas ocorrências frequentemente ocorrem na estrutura da comunidade de projetos de software de código aberto(open-source)[10], tal fato se dá pela maior

presença de trabalhos na literatura usando dados de projetos de código aberto do que de código fechado.

A fim de ilustrar de que não há consenso, outro estudo na literatura intitulado de “Splicing Community and Software Architecture Smells in Agile Teams: An industrial Study”[9] avaliando os *community smells* dentro do cenário de times ágeis atesta que os *smells* que ocorrem com mais frequência são :

- Time-warp;
- Distância Cognitiva;
- DevOps Clash;

O fato de não ter uma anuência dentro da comunidade científica acerca do questionamento é refletida em outros estudos na área, como no caso do artigo “Gender Diversity and Community Smells: Insights From the Trenches”, o qual afirma que embora os *community smells* sejam cada vez mais estudados, pouco se sabe sobre como é a composição da equipe, incluindo até a diversidade de gênero[7], reforçando a necessidade de se compreender as equipes de software a fim de evitar os *community smells*.

Além do mais, há observações constatadas, como afirma o artigo “Impacts of software community patterns on process and product: An empirical study”, de que a ocorrência de específicos *community smells* em combinação com tipos de padrões de comunidade específicos leva à conjectura de que existem padrões de comunidade mais eficazes, por meio dos quais os *community smells* específicos da comunidade podem ser mitigados[17].

## 5. CONSIDERAÇÕES FINAIS

Neste artigo, realizamos um estudo de mapeamento sistemático da literatura acerca dos *community smells*. Posteriormente, foi possível selecionar artigos que abordam o assunto e catalogá-los para compreender como o estado da arte vem sendo discutido. Foram analisados 24 estudos no total sobre *community smells*.

Visando alcançar esse objetivo foi necessário, primeiramente, a elaboração de questões que orientaram o trabalho presente com o objetivo de compreender o tema. Primeiramente, neste artigo,

exploramos o estado da arte a fim de analisar o quão consolidado se encontra as ferramentas de detecção de *community smells*.

Embora existam ferramentas de detecção, sua amplitude e cobertura para captar os tipos de *community smells* ainda está em desenvolvimento, sendo tímida se comparada ao vasto leque de *community smells* existentes.

Com poucos estudos para definir, caracterizar e detectar os *community smells* em projetos, eles são aplicados, em geral, a um conjunto limitado de projetos, e sua generalização requer esforço manual e experiência humana para definir e calibrar um conjunto de regras de detecção para combinar os sintomas de um *community smells* com as características reais de um determinado projeto de software [13,19].

Aliás, por ainda ser um tema que ainda está escasso e na sua infância, as discussões se apresentam em poucas publicações complexas como em artigos de periódicos. Além do mais, a literatura atualmente se concentra em um pequeno número de *community smells*.

Outrossim, nosso segundo questionamento de pesquisa expôs a carência de diversidade de estudos mais aprofundados acerca da aparição frequente de *community smells* nas equipes de software, estudos os quais no futuro poderia servir para promover a descoberta de padrões de comunidades mais adequados para projetos tanto de código aberto, quanto de código fechado.

Portanto, sendo difícil obter uma compreensão consistente das características dos mais diversificados *community smells*, em que as informações existentes estão dispersas nos poucos estudos disponíveis.

## 6. REFERENCIAS

[1] ALBERTIN BONO MILAN, LUIZ FERNANDO. Adoção do modelo aberto de desenvolvimento de software pelas Empresas. 2018. 99 p. Tese de Doutorado — Fundação Getulio Vargas, [s. l.], 2018. Disponível em: [https://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/20689/Tese\\_Luiz\\_Fernando\\_Albertin\\_Bono\\_Milan.pdf?sequence=1&isAllowed=y](https://bibliotecadigital.fgv.br/dspace/bitstream/handle/10438/20689/Tese_Luiz_Fernando_Albertin_Bono_Milan.pdf?sequence=1&isAllowed=y). Acesso em: 13 Maio 2022.

- [2] DITTRICH, Yvonne; NØRBJERG, Jacob; TELL, Paolo; BENDIX, Lars. Researching Cooperation and Communication in Continuous Software Engineering. 2018 IEEE/ACM 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), p. 87-90, 2018.
- [3] ALAM KHAN, Mohammed Shafiu; AHMED, Sumon; AHAMMED, Toukir. Do Missing Link Community Smell Affect Developers Productivity: An Empirical Study. Knowledge Engineering and Data Science, [S. l.], v. 4, n. 1, p. 29-37, jul. 2021. DOI 10.17977/um018v4i12021p29-37. Disponível em: <http://journal2.um.ac.id/index.php/keds/article/view/21527>. Acesso em: 13 dez. 2022.
- [4] FLOWER, Martin. Refactoring: Improving the Design of Existing Code. [S. l.]: Addison-Wesley, 1999.
- [5] TAMBURRI, Damian A.; KRUCHTEN, Philippe; LAGO, Patricia; VLIET, Hans van. Social debt in software engineering: insights from industry. Journal of Internet Services and Applications, [S. l.], p. 1-17, 2015. DOI 10.1186/s13174-015-0024-6. Disponível em: [https://www.researchgate.net/publication/277355779\\_Social\\_debt\\_in\\_software\\_engineering\\_insights\\_from\\_industry](https://www.researchgate.net/publication/277355779_Social_debt_in_software_engineering_insights_from_industry). Acesso em: 19 jun. 2022.
- [6] A. TAMBURRI, Damian; KRUCHTEN, Philippe; LAGO, Patricia; VAN VLIET, Hans. What is social debt in software engineering?. International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, ano 2013, n. 6, p. 1-4, 25 maio 2013. DOI 10.1109/CHASE.2013.6614739. Disponível em: <https://ieeexplore.ieee.org/document/6614739>. Acesso em: 13 ago. 2022.
- [7] CATOLINO, Gemma; PALOMBA, Fabio; A. TAMBURRI, Damian; SEREBRENIK, Alexander; FERRUCCI, Filomena. Gender Diversity and Community Smells: Insights From the Trenches. IEEE, [S. l.], v. 37, p. 1-7, 30 nov. 2019. DOI 10.1109/MS.2019.2944594. Disponível em: <https://ieeexplore.ieee.org/document/8852638/>. Acesso em: 14 ago. 2022.
- [8] KITCHENHAM, Barbara; CHARLES, Stuart. Guidelines for performing Systematic Literature Reviews in Software Engineering. Keele University and Durham University Joint Report, [S. l.], v. 2, p. 1-66, 9 jul. 2007.
- [9] TAMBURRI, Damian A.; KAZMAN, Rick; VAN DEN HEUVEL, Willem-Jan. Splicing Community and Software Architecture Smells in Agile Teams: An industrial Study. Proceedings of the 52nd Hawaii International Conference on System Sciences, [S. l.], p. 7037-7047, 8 jan. 2019. DOI 10.24251/HICSS.2019.843. Disponível em: <https://hdl.handle.net/10125/60140>. Acesso em: 15 dez. 2022
- [10] MUMTAZ, Haris; SINGH, Paramvir; BLINCOE, Kelly. Analyzing the Relationship between Community and Design Smells in Open-Source Software Projects: An Empirical Study. ESEM '22: Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, [S. l.], p. 23-33, 19 set. 2022. DOI 10.1145/3544902.3546249. Disponível em: <https://dl.acm.org/doi/10.1145/3544902.3546249>. Acesso em: 12 dez. 2022.
- [11] SCHNEIDER, Michael. Costs of poor communication reach \$37 billion. Avoid disconnects by implementing these 2 things. Inc.com, 11 jul. 2018. Disponível em: <https://www.inc.com/michael-schneider/the-extrovert-vs-introvert-dynamic-could-be-costing-your-organization-millions-heres-how-to-bridge-communication-gap.html>. Acesso em: 18 dez. 2022.
- [12] CATOLINO, Gemma; PALOMBA, Fabio; TAMBURRI, Damian A.; SEREBRENIK, Alexander; FERRUCCI, Filomena. Gender Diversity and Women in Software Teams: How Do They Affect Community Smells?. 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), [S. l.], p. 11-20, 2019. DOI 10.1109/ICSE-SEIS.2019.00010. Disponível em: <https://ieeexplore.ieee.org/document/8797636>. Acesso em: 13 dez. 2022.
- [13] FONTANA, Francesca Arcelli; TAMBURRI, Damian Andrew; OLIVETO, Rocco; ZAIDMAN, Andy; SEREBRENIK, Alexander; PALOMBA, Fabio. Beyond Technical Aspects: How Do Community Smells Influence the Intensity of Code Smells?. IEEE Transactions on Software Engineering, [S. l.], v. 47, n. 1, p. 108 - 129, 27 nov. 2018. DOI 10.1109/TSE.2018.2883603. Disponível em: <https://ieeexplore.ieee.org/document/8546762/>. Acesso em: 13 dez. 2022.
- [14] CATOLINO, Gemma; PALOMBA, Fabio; TAMBURRI, Damian A.; SEREBRENIK, Alexander; FERRUCCI, Filomena. Refactoring Community Smells in the Wild: The Practitioner's Field Manual. 2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), p. 25-34, 2020.
- [15] STEFANO, Manuel de; PECORELLI, Fabiano; TAMBURRI, Damian Andrew; PALOMBA, Fabio; LUCIA, Andrea de. Splicing Community Patterns and Smells: A Preliminary Study. In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20), p. 703-710, 25 set. 2020. DOI 10.1145/3387940.3392204. Disponível em: <https://dl.acm.org/doi/10.1145/3387940.3392204>. Acesso em: 13 dez. 2022.
- [16] CATOLINO, Gemma; PALOMBA, Fabio; TAMBURRI, Damian A.; SEREBRENIK, Alexander. Understanding Community Smells Variability: A Statistical Approach. 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Society, [S. l.], p. 77-86, 2021. DOI 10.1109/ICSE-SEIS52602.2021.00017. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9402214>. Acesso em: 18 jan. 2023.
- [17] STEFANO, Manuel De; IANNONE, Emanuele; PECORELLI, Fabiano; TAMBURRI, Damian Andrew. Impacts of software community patterns on process and product: An empirical study. Science of Computer Programming, v. 214, 2022. DOI 10.1016/j.scico.2021.102731. Disponível em:

<https://www.sciencedirect.com/science/article/abs/pii/S0167642321001246>. Acesso em: 13 dez. 2022.

[18] ALMARIMI , Nuri; MKAOUER , Mohamed Wiem; SAIDANI, Islem; CHOUCHEM, Moataz; OUNI, Ali. On the detection of community smells using genetic programming-based ensemble classifier chain. Proceedings of the 15th International Conference on Global Software Engineering, [S. l.], p. 43-54, jun. 2020. DOI <https://doi.org/10.1145/3372787.3390439>. Disponível em: <https://dl.acm.org/doi/10.1145/3372787.3390439>. Acesso em: 23 dez. 2022.

[19] TAMBURRI, Damian A.; PALOMBA, Fabio; KAZMAN, Rick. Exploring Community Smells in Open-Source: An Automated Approach. IEEE Transactions on Software Engineering, [S. l.], v. 47, n. 3, p. 630 - 652, 24 fev. 2019. DOI 10.1109/TSE.2019.2901490. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8651329>. Acesso em: 23 dez. 2022.

[20] TAMBURRI, Damian Andrew; PALOMBA, Fabio; SEREBRENIK, Alexander; ZAIDMAN, Andy. Discovering community patterns in open-source: a systematic approach and its evaluation. Empirical Software Engineering, [S. l.], v. 24, n. 3, p. 1369–1417, jun. 2019. DOI [doi.org/10.1007/s10664-018-9659-9](https://doi.org/10.1007/s10664-018-9659-9). Disponível em: <https://dl.acm.org/doi/abs/10.1007/s10664-018-9659-9>. Acesso em: 23 dez. 2022.

[21] MUMTAZ, Haris; PARADIS, Carlos; PALOMBA, Fabio; TAMBURRI, Damian A.; KAZMAN, Rick; BLINCOE, Kelly. A Preliminary Study on the Assignment of GitHub Issues to Issue Commenters and the Relationship with Social Smells. 2022 IEEE/ACM 15th International Workshop on Cooperative and Human Aspects of Software Engineering , IEEE, p. 61-65, 2022. DOI 10.1145/3528579.3529181. Disponível em: <https://ieeexplore.ieee.org/document/9796373>. Acesso em: 20 jan. 2023.

[22] EKEN, Beyza; PALMA, Francis; AYŞE , Başar; AYŞE , Tosun. An empirical study on the effect of community smells on bug prediction. Software Quality Journal volume 29, [S. l.], p. 159–194, 15 fev. 2021. DOI [doi.org/10.1007/s11219-020-09538-7](https://doi.org/10.1007/s11219-020-09538-7). Acesso em: 19 jan. 2023.

[23] HUANG , Zi-Jie; SHAO, Zhi-Qing; FAN, Gui-Sheng; YU, Hui-Qun; YANG, Xing-Guang; YANG, Kang. Community Smell Occurrence Prediction on Multi-Granularity by Developer-Oriented Features and Process Metrics. Journal of Computer Science and Technology, [S. l.], v. 37, n. 1, p. 182–206, 1 jan. 2022. DOI 10.1007/s11390-021-1596-1. Disponível em: <https://dl.acm.org/doi/abs/10.1007/s11390-021-1596-1>. Acesso em: 20 jan. 2023.

[24] ALMARIMI, Nuri; OUNI , Ali; CHOUCHEM, Moataz; MKAOUER, Mohamed Wiem. CsDetector: an open source tool for community smells detection. ESEC/FSE 2021: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, [S. l.], p. 1560–1564, 18 ago. 2021. DOI <https://doi.org/10.1145/3468264.3473121>. Disponível em:

<https://dl.acm.org/doi/10.1145/3468264.3473121>. Acesso em: 18 jan. 2023.

[25] SARMENTO, Camila; MASSONI, Tiago; SEREBRENIK, Alexander; CATOLINO, Gemma; TAMBURRI, Damian; PALOMBA, Fabio. Gender Diversity and Community Smells: A Double-Replication Study on Brazilian Software Teams. 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering , [S. l.], p. 273-283, 2022. DOI 10.1109/SANER53432.2022.00043. Disponível em: <https://ieeexplore.ieee.org/document/9825896>. Acesso em: 18 jan. 2023.

[26] LAMBIASE, Stefano; CATOLINO, Gemma; TAMBURRI, Damian A.; SEREBRENIK, Alexander; PALOMBA, Fabio; FERRUCCI, Filomena. Good Fences Make Good Neighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells. 2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Society , [S. l.], p. 67-78, 2022. DOI 10.1145/3510458.3513015. Disponível em: <https://ieeexplore.ieee.org/document/9793992/>. Acesso em: 17 jan. 2023.

[27] PALOMBA, Fabio; TAMBURRI, Damian A.; SEREBRENIK, Alexander; ZAIDMAN, Andy; FONTANA , Francesca Arcelli; OLIVETO, Rocco. How do community smells influence code smells?. ICSE '18: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings, [S. l.], p. 240–241, 27 maio 2018. DOI <https://doi.org/10.1145/3183440.3194950>. Disponível em: <https://dl.acm.org/doi/10.1145/3183440.3194950>. Acesso em: 18 jan. 2023.

[28] ALMARIMI, Nuri; OUNI, Ali; MKAOUER, Mohamed Wiem. Learning to detect community smells in open source software projects. Knowledge-Based Systems , [S. l.], v. 204, 2020. DOI <https://doi.org/10.1016/j.knosys.2020.106201>. Acesso em: 17 jan. 2023.

[29]HUANG, Zijie; SHAO, Zhiqing; FAN, Guisheng; GAO, Jianhua; ZHOU, Ziyi; YANG, Kang; YANG, Xingguang. Predicting Community Smells' Occurrence on Individual Developers by Sentiments. 2021 IEEE/ACM 29th International Conference on Program Comprehension, [S. l.], p. 230-241, 2021. DOI 10.1109/ICPC52881.2021.00030. Acesso em: 17 jan. 2023.

[30] PALOMBA, Fabio; TAMBURRI, Damian Andrew. Predicting the emergence of community smells using socio-technical metrics: A machine-learning approach. Journal of Systems and Software, [S. l.], v. 171, p. 20-37, 2021. DOI <https://doi.org/10.1016/j.jss.2020.110847>. Acesso em: 17 jan. 2023.

[31] TAMBURRI, Damian A. Software Architecture Social Debt: Managing the Incommunicability Factor. IEEE Transactions on Computational Social Systems, [S. l.], v. 6, n. 1, p. 20 - 37, 15 jan. 2019. DOI 10.1109/TCSS.2018.2886433. Disponível em: <https://ieeexplore.ieee.org/document/8613009>. Acesso em: 19 jan. 2023.