Received: 6 January 2022 | Revised: 27 September 2023 | Accepted: 24 October 2023

DOI: 10.1111/isj.12485

RESEARCH ARTICLE

WILEY

An empirical investigation of social comparison and open source community health

Kevin Lumbard[1] | Matt Germonprez[2] | Sean Goggins[3]

[1]Department of Computer Science, Design & Journalism, Creighton University, Omaha, Nebraska, USA

[2]College of Information Science &Technology, University of Nebraska at Omaha, Omaha, Nebraska, USA

[3]College of Engineering, University of Missouri, Columbia, Missouri, USA

Correspondence
Kevin Lumbard, Department of Computer Science, Design & Journalism, Creighton University, Omaha NE, USA.
Email: kevinlumbard@creighton.edu

Abstract

It is well known that corporations rely on open source software as part of their product development lifecycle. Given these commitments, understanding the health of open source communities is a central concern in today's business setting. Our research uses social comparison theory as a framework for understanding how open source communities consider community health beyond any single metric within any single open source community—including a broader view of how others are using these health indicators in practice. Using methods from engaged field research, including 38 interviews, we examine practices of social comparison as an advancement in understanding open source community health—and subsequently engagement with open source communities. The results of this study show that open source community health is not a single set of discrete metrics but is an ongoing social construction. Through our study, we advance theoretical and applied knowledge regarding issues of open source community health, open source community engagement, and social comparison.

KEYWORDS

community health, field research, open source, social comparison

1 | INTRODUCTION

Modern digital infrastructure is built with and relies on the development of open source software (Eghbal, 2016). Open source communities, the communities that develop open source software, enable people and organizations to build and maintain complex software and manage relationships between software components (Germonprez et al., 2018; Howison & Crowston, 2014; Lindberg et al., 2016). Open source is a unique social construction that is experiencing ongoing and significant shifts in its relationship to technology work (Germonprez et al., 2018). The growth of open source software communities responsible for building and maintaining complex and intertwined pieces of strategic infrastructure makes awareness of open source health vital for the ongoing success of each participating member. If even just a single open source community dies, it can have cascading impacts on downstream projects that rely on its outputs.

Open source community health is a complex topic, and existing research provides an important starting point; however, existing research has focused primarily on individual communities or projects which provides little guidance for open source practitioners who are concerned with how their work and interests affect their organizations and interdependent communities in open source ecosystems. Our research extends the scale and scope of open source community health to include the relationships between people, projects, and organizations in open source ecosystems (Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021). Extending the scope of open source community health to many different communities allows us to explore health indicators from different perspectives and contexts enabling our first contribution: *Exploring how open source communities understand and define open source community health*.

Investigations of single open source communities, such as the Linux kernel or Kubernetes, have yielded fruitful research contributions. Yet, open source communities rarely exist in isolation (Thomas & Hunt, 2004). They are situated in relation to other open source communities through upstream dependencies, complementary services, and shared community members. A look at the Cloud Native Computing Foundation Landscape[1] illustrates projects in relation to others, particularly around complementary services. Considering open source communities relative to others, as we do in this paper, certainly provides a shallower look within a single open source community but provides a deeper look at the interactions present across open source communities (Germonprez et al., 2017; Li et al., 2022). Such a position enables our second expected contribution in this paper: *Examining interactions present across open source communities assists how we understand open source, not as a single community but as a collection of communities*.

Revealing interactions present across open source communities could start from several points including licence management (Singh & Phelps, 2013), dependency management (Gustavsson, 2020), and community governance (O'Mahony, 2007). Licences play an important role in how open source software is legally distributed and used, carrying risk in downstream compliance failures. Dependencies play a key role in how open source software is technically distributed and used, creating complex networks of open source software. Community governance is now steered by board members, representing project, corporate, and foundation members. In each of these starting points, research could begin at a community level but quickly come to include a narrative that comes to include a constellation of open source communities (Zhang et al., 2014).

To these points, our research reveals interactions present across open source communities through open source community health. Open source community health includes concerns of licensing, dependencies, and governance as a collection of indicators of a community's ability to continue to produce quality artefacts (Germonprez et al., 2018; Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021; Naparat et al., 2015). OpenSSL was one of the highest profile cases of an open source community failure having cascading, downstream effects on projects that relied on the OpenSSL software (Durumeric et al., 2014). The failure of critical open source communities can result in the implementation of crisis management mechanisms for organizations, including sudden, increased investment of organizational time, or abandonment of open source communities in crisis. Given these risks, corporations are interested in proactively committing resources to support critical communities, ensuring open source ecosystem stability.

Indicators of open source community health can include (1) development activity as an expression of community success and sustainability (Crowston et al., 2006; Crowston & Howison, 2006, 2011), (2) community activity as an expression of open source software release cadence (Jiang et al., 2014; Schweik & English, 2012), and (3) social

[1] https://landscape.cncf.io/.

activity as an expression of community culture and social behaviour (Daniel & Stewart, 2016; Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021). Much of this prior research leans into health indicators as effective within individual communities, highlighting open source communities as community-based social constructions (Crowston & Howison, 2011). While certainly true, our research leans out and explores open source health indicators as effective across communities, highlighting open source communities as constructions that extend beyond any single community. This enables our third expected contribution in this paper: *Open source community health provides a vehicle for exploring and comparing the different contexts that exist for open source communities*.

To support our first three expected contributions, we apply Festinger's social comparison theory (Festinger, 1954) as a framework for exploring how open source community health can be used to understand open source, not as a single project but as a collection of projects. Social comparison is a relevant framework for exploring open source community interactions because it explains how appraisal and evaluation of abilities are formed through comparison between social groups (Festinger, 1954). Our use of Festinger et al. (1954) enables evaluations of open source community health to be operationalized through comparisons across open source communities, and not only the applications of health indicators within a single open source community.

Centring our study on social comparison highlights themes observed across communities and sheds light on the increasingly complex ways that project context influences interpretations of open source communities. As organizations orient open source work, we can explore how perceptions can change through comparisons (Festinger, 1954; Tajfel, 1974; Tajfel & Turner, 1978). Throughout our research, we investigate open source community health as more than just any one discrete indicator (i.e., number of issues, age of a pull request, or volume of bot activity) and investigate open source community health as situated in a larger social setting where people not only examine discrete indicators within a community but compare those indicators across communities. In doing so, we advance our understanding of open source community research through acts of social comparison and advance knowledge about how corporations view their increasingly essential relationships with open source communities. This enables our fourth and final expected contribution in this paper: *Social comparison theory supports people's use of community health indicators to develop open source as a construction beyond any single community*.

2 | OPEN SOURCE AS A CONSTRUCTION BEYOND ANY SINGLE COMMUNITY

Early references to open source in Information Systems began in 2000 with papers published at ECIS, AMCIS, and ICIS conferences. In these earliest papers, Ljungberg (2000) and Sawyer (2000) focused on open source as a new model for organizing, Hars and Ou (2000) asked how open source can be a viable software development option, and Feller and Fitzgerald (2000) provided on a structure for researchers to examine open source software development. In these, the most common open source communities referenced were the Linux kernel, the Apache Web server, and Netscape Navigator. None of this early research included a precise investigation of open source communities; instead, the communities provided a backdrop for speaking to the emerging phenomena of open source.

Through the early 2000s, Information Systems Research began to investigate open source communities more directly. This included why people participate in open source communities concerning network, gifting, and intrinsic motivations of contributors. Looking more precisely at open source communities, this research began to explore measures of trust, success, leadership, satisfaction, coordination, governance, and control (Chengalur-Smith & Sidorova, 2003; Crowston et al., 2003; Stewart & Ammeter, 2002; Stewart & Gosain, 2001). Additionally, researchers began to investigate open source in relation to the commercial software industry (Feller & Fitzgerald, 2003; Heikinheimo & Kuusisto, 2004; Lerner & Tirole, 2002), focusing on how these two paradigms could be complementary, not necessarily alternatives to each other. During this time, research highlighted the (1) dynamics within individual open source communities and (2) how open source was connected with corporate modes of production, including preliminary recognition of open source as a construction beyond any single project (Thomas & Hunt, 2004).

With the growing research on open source communities, several frameworks were published to support and guide this work (Aksulu & Wade, 2010; Fitzgerald, 2006; Nelson et al., 2006). These frameworks helped to organize work and orient researchers to the new insights related to open source communities that include discrete measures along with the larger impacts that open source can have, for example, on the software development industry (Walia et al., 2006). As seen from these frameworks, research on corporate engagement with open source started to become common articulating business models for open source engagement (Riehle, 2009) and the adoption of open source software within a firm (Bhadauria et al., 2009; Chengalur-Smith et al., 2010; Feller et al., 2008; Lundell et al., 2008; Mehra et al., 2011). While not precisely looking at single open source communities in relation to other projects, this corporate-communal work highlighted the reality that open source communities exist in complex systems of people and technology.

During this time, research often placed single open source communities in relation to other communities (Hu & Zhao, 2008; Scacchi et al., 2006), focusing on the networks present across open source communities (Hahn et al., 2008). As discrete measures within single open source communities remained prevalent (Faraj et al., 2015; Lucia Kim & Teo, 2013; Shaikh & Cornford, 2012; Torkar et al., 2011), research highlighted network effects present in open source (Singh et al., 2011)—strengthening insight that open source is more than just a single community (Chou & He, 2011). Important work expressed that open source can be understood as a collection of projects, focusing on distributed skill development (Mehra & Mookerjee, 2012; von Krogh et al., 2012), resource distribution across open source communities (Walsh et al., 2012), networks of firms engaging with open source (Han et al., 2012), knowledge reuse between communities (Zargar, 2013), and open source as a constellation of projects (Zhang et al., 2014).

Research now regularly presents open source as a distributed and networked engagement, highlighting the impacts of diversity on global and distributed open source collectives (Daniel et al., 2013), how innovation is a collective and networked endeavour (Germonprez et al., 2019; Liu et al., 2017), and how network externalities play a role in open source product development (Germonprez et al., 2017; Machado et al., 2017). As a collection of communities, open source is now understood as a phenomenon of software ecosystems (Franco-Bedoya et al., 2017; Teixeira et al., 2016; Zargar, 2013) that have come to include corporations competing and collaborating within this space (Germonprez et al., 2020; Singh & Phelps, 2013; Wynn et al., 2008). Drawing forward the reality that open source communities are often directly connected or influenced by network externalities, work like Xiao et al. (2018, p. 1242) helps us realize open source as an "interrelated Webs of developers, technologies and projects—rather than distinct projects or tools."

3 | OPEN SOURCE COMMUNITY HEALTH: A CORPORATE-COMMUNAL PERSPECTIVE

Corporate engagement with open source is opening new points of connection between research and practice, particularly regarding how open source community health is evaluated (Germonprez et al., 2018). Increasing corporate engagement and the need to maintain ecosystems of interdependent software for strategic needs highlights the expectation of open source projects to continue to serve the needs of community members as well as downstream software users (Arantes & Freire, 2011; Bietz et al., 2012; Chengalur-Smith et al., 2010; Gamalielsson & Lundell, 2014; Nyman & Lindman, 2013; Schweik & English, 2012). As such, understanding open source community health as an expression of longevity requires health indicators that can inform a community's likelihood or ability to continue to produce quality software for all (Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021; Link & Germonprez, 2018; Naparat et al., 2015).

3.1 | Measures of open source community health

The quality of open source software is visible through indicators of the coordination mechanisms used in software development (Jiang et al., 2014). For example, community activity measures centre on the frequency of open source software releases (Jiang et al., 2014). Releases are a common milestone based on the premise that each release adds desired features and increases stability (Sharma et al., 2015). Each release signals some advancement towards feature completeness, with more frequent releases implying higher development velocity. Measures of activity stand as one measure of health that is complemented by indicators of software quality and community process, including issue management, code commits, and coordinating activities (Blincoe et al., 2012; Crowston et al., 2006; Crowston & Howison, 2011).

Research to recognize open source community health focused on success, defined as an estimate of the likelihood "my project will take off." In this context, success requires project activity and the release of code, without which a project may be abandoned (Schweik & English, 2012). The use of open source software as a component in larger systems motivates external success measures (Crowston et al., 2003), emphasizing activity measures specifically. Higher levels of activity are associated with project growth, output, development process, and the outcomes for community members (Aué et al., 2016; Crowston et al., 2006; Crowston & Scozzi, 2008; Krishnamurthy, 2002; Lee et al., 2009; Subramaniam et al., 2009). Activity measures are, therefore, a common proxy for several distinct signals of success. In particular, prior research focuses on investigations of metrics for code, diversity, organizational dynamics, risk, ecosystem state, adverse event prediction, and end-user tools (see Table 1).

These indicators of community health each reflect specific aspects of an open source community, each of which informs our understanding of a project's ability to produce quality software. Open source community health research has advanced to include context and consideration of how social interaction represents a community's state. Thus, building an understanding of community health highlights the importance of specific concerns, like social dynamics and community diversity (Aué et al., 2016; Daniel et al., 2013; Daniel & Stewart, 2016; Naparat et al., 2015; Terrell et al., 2017). For example, members join communities and at first become socialized in a community's context. Socialization often involves an orientation such as starting as a user, becoming a contributor, and, in some cases, assuming leadership responsibilities as a maintainer (Jergensen et al., 2011). This is often referred to as the contributor funnel or the path to leadership. Health, in such cases, may be expressed in ratios of successful socialization, such as new contributors as a proportion to contributors who leave (Crowston & Howison, 2011).

TABLE 1 Open source community health activity metrics focus.

Metric	Explanation	References
Code	Procedures and artefacts associated with an open source software code base	Davies et al. (2011); Izquierdo-Cortazar et al. (2017); Johari & Kaur (2012); Mondal et al. (2012); Rahman & Devanbu (2013); Sharma et al. (2015)
Diversity	The inclusion of all people and organizations with an interest in the development of open source artefacts	Aué et al. (2016); Daniel et al. (2013)
Organizational dynamics	Characteristics of how organizations work internally and collaboratively in the development of open source artefacts	Capiluppi et al. (2012); Meneely et al. (2014); Wang & Lantzy (2011); Yamashita et al. (2015); Youssef & Capiluppi (2015)
Risk	Evident security concerns and liabilities associated with the development of open source and artefacts	Johari & Kaur (2011); Kaur & Kaur (2012); Madeyski & Kawalerowicz (2017)
Ecosystem state	Dependencies between open source software projects as part of a larger collection of projects	Jansen (2014); Ortu et al. (2015); Plakidas et al. (2016); Yamashita et al. (2015)
Adverse event prediction	Anticipation of failures in an open source project or open source ecosystem	Abunadi & Alenezi (2015); Madeyski & Kawalerowicz (2017); Rausch et al. (2017)
End-user tools	Determination of downstream use of open source artefacts	Hannemann et al. (2014); Jansen (2014); Rosen et al. (2015); Rozenberg et al. (2016)

This prior work provides a necessary foundation for understanding open source community health. However, this prior research is primarily localized to individual communities and does not provide insight into the health of across communities (Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021). Open source communities exist within software ecosystems, evolve, and include many different people and organizations (Germonprez et al., 2017). Localized indicators of health do not always reveal the broader workings of open source communities and the ecosystems that they occupy (Germonprez et al., 2018). From a local perspective, findings and theories of open source communities produce specific knowledge with claims constrained to that context. However, community members have questions like "will a key open source community survive?" and "how can we contribute to the success of a critical open source community?" Providing answers to these questions is complicated because communities are partly social endeavours that perplex assumptions of project uniformity implied by referencing individual indicators. Open source community health can be better understood when a community's context, including accompanying assumptions, entails more context than localized community activity counting alone can provide.

3.2 | Open source community health beyond any single community

Our research builds on the published efforts that have demonstrated the role and power of different indicators of open source community health. In our research, we look past any single metric (Muller, 2019) to include the ways that people draw indicators together in the construction of their own, often complex, visions of open source community health. We contend that open source community health is not only the rise and fall of certain indicators but also how that rising and falling is understood in context across communities. Thus, open source communities define community health within software ecosystems and that definition may fluctuate with time and contextual shifts. Curto-Millet and Shaikh (2017) showed that *openness* in open source communities is itself a dynamic concept that has multiple and co-existing meanings that cause open source communities to continuously reflect on the status of their projects. Similarly, open source community health is a continuous reflection by community members, which is made more difficult to quantify in an era where corporate engagement is a rapidly increasing factor in open source ecosystems (Germonprez et al., 2018) and communities themselves exist in relation to others (Xiao et al., 2018). To explore this further, we introduce social comparison (Festinger, 1954) as our way to understand how open source community health is constructed by community members, providing a way to strengthen our understanding of complexities inherent in applying a collection of largely quantified health indicators across the increasingly complex software ecosystems found in open source today.

4 | SOCIAL COMPARISON

Social comparison explains how the appraisal and evaluation of abilities are formed through comparison in social groups (Festinger, 1954). People are driven to evaluate their own abilities and, in the absence of an objective measure, will compare themselves with others rather than persist in the cognitive dissonance of not having an objective score (Festinger, 1954). Dissonance drives people to seek comparisons with individuals most like themselves. With a hint of irony, the human need to reconcile dissonance drives resistance to quantitative, objective measures when the skills, abilities, or even health being evaluated are perceived as too complex to evaluate (Buckingham & Alicke, 2002).

Festinger's theory of social comparison (Festinger, 1954) emerges from four separate and distinct prior studies (Dreyer, 1954; Festinger, 1950; Festinger et al., 1954; Hoffman et al., 1954). Festinger (1950) is a synthesis of experiments and field studies examining motivations and effects of informal social communication. Festinger et al. (1954) centre on experiments that explore the cause and effect of self-evaluation in relation to group membership. Dreyer (1954) conducts an experiment with sixth graders using a card sorting technique, demonstrating that cognitive factors are determinants of aspiration level, and Hoffman et al. (1954) demonstrate the influence of group membership in coalitions using experiments.

As individuals, the tensions between measurements and how we see ourselves shape our identities. For example, our own performance expectations will rise or fall with what we believe we can accomplish in relation to others (Bandura, 1977). Yet, we often must make our social comparisons when information is not clear, effectively reinforcing or dislodging our sense of self-efficacy. Individual identity interacts with our social context to resolve the dissonance felt between our identity and the available social comparisons. In some ways, this dynamic aids learning, which advances open source community health as we learn from people whose perspectives, and abilities are different from our own (Vygotsky, 1964).

4.1 | Identity, evolution, and comparison

Identity development also interacts with social comparison in the ways we seek out other individuals who are more like us and reinforce our identities with less dissonance. Who we are is also defined by the social groups we identify with, and we tend to find our most outstanding comfort with the people we perceive to be most like us (Tajfel & Turner, 1978; Turner et al., 1979). Open source community members, being human, follow the same social psychological guideposts. Once we identify with and are accepted in a group, like an open source community, we begin the process of comparison all over again (Tajfel, 1974; Turner et al., 1979). Sometimes, we wear the jerseys of our favourite athletic teams or join online communities that reinforce our identities (Ren et al., 2007). In open source software, this reinforcement occurs in part through community engagement.

Communities are more than containers for identity reinforcement. They also require task completion, self-care, and connections with other sets of people and institutions (McGrath, 1984), and none of the processes that underlie community development are linear. The communities we are part of go through stages in practice, from formation to operation and ultimately metamorphosis (Arrow et al., 2000; McGrath, 2000). Community formation and metamorphosis include cycles of punctuated equilibrium (Gersick, 1988), where the "punctuation" (i.e., disruption) is driven by events that convince the group it must achieve a higher standard or shift its goals entirely in response to external events. As such, when measuring performance, community members seek comparable communities using the objective measures and rankings most salient in the moment in the different stages of the communities they are part of (Cummings & Cross, 2003; McGrath, 1991).

4.2 | Social comparison and open source community health

Open source communities arise from the contributions and work of people. Open source communities are social networks and can be grouped by their common and varying characteristics (Gacek & Arief, 2004). Researchers and practitioners often compare common characteristics using objective measures across many projects to understand the dynamics of open source communities. Open source collaboration platforms such as SourceForge, GitLab, Bitbucket, and GitHub make comparisons and analysis of projects and collections of projects possible (Aué et al., 2016; Chengalur-Smith et al., 2010; Crowston et al., 2006; Daniel et al., 2013; Raja & Tretter, 2012).

Prior research explores the context within which open source communities evolve, grouping and sorting them along measures like popularity (Crowston & Howison, 2006), age (Chengalur-Smith et al., 2010), community size (Chengalur-Smith et al., 2010; Crowston et al., 2006), collaboration methods (Daniel & Stewart, 2016), and diversity (Aué et al., 2016; Daniel et al., 2013). Prior research leaves room to solve problems associated with understanding how these measures are related to the health of open source communities and limited guidance for open source participants on how to assess the health of their individual communities. Social comparison (Festinger, 1954) enables us to connect with open source community members directly to see how they construct open source community health based on comparisons of communities they want to be like and communities they want to avoid. Social comparison draws the human element forward in determining how open source community health is constructed with reference to people's surroundings, particularly how comparisons are used in the ongoing evaluation of open source community health and, subsequently, organizational engagement with open source communities. The original social comparison studies took place more than a half-century ago (Dreyer, 1954; Festinger, 1950; Festinger et al., 1954; Hoffman et al., 1954), in social and societal contexts that are substantially different than the present day—a day that is significantly more mediated by technology.

5 | RESEARCH DESIGN

Our research programme primarily explores open source communities hosted by the Linux Foundation,[2] one of the largest open source foundations—nurturing organizational engagement with over 900 open source communities. The Linux Foundation supports critically important open source communities and is financially backed by leading technology companies, including Google, AT&T, Cisco, IBM, Intel, Meta, and Microsoft. Members of our research team have been working with the Linux Foundation and associated projects for the past twelve years, dating back to when the Linux Foundation was a considerably smaller enterprise, supporting only eight projects. At a general level, our twelve-year field data includes over 250 interviews, 1500 survey responses, 20 focus groups, and 2000 pages of field notes. Across all our studies during this time, our methods have included field engagement, participant observation (Spradley, 2016a), group informatics, which includes trace data analysis, social network analysis, qualitative coding, computational linguistic analysis (Goggins et al., 2013), and digital ethnography (Geiger & Ribes, 2011; Kozinets, 2015). While this previous work provides us with insight into open source work resulting from using several different methodologies, the current research is a purely qualitative field study and is focused on building an understanding of open source community health by conducting interviews with open source practitioners.

Our current research is pragmatic and based on the proposition that researchers should use the methodological approach that works best for the research problem that is being investigated (Tashakkori & Teddie, 1998). In our current research, we take an interpretivist stance for our data collection and analysis. Throughout this process, we (1) built an understanding of concepts that have been explored in previous literature, (2) developed new concepts and understanding within our research context, and (3) deepened our understanding of these prior and new concepts in open source practice. The current research explores the perceptions of community members about open source health and what indicators they believe are important for their contexts. To narratively explore these perceptions, we interviewed and discussed open source community health metrics with 38 participants. To be clear, our discussions with people about open source community health metrics did not lead us to use any quantitative data about specific metrics discussed by interviewees. In this current research, we do not perform any quantitative analysis or make any quantitative assertions about such things as the probability or predictability of open source community health metrics.

5.1 | Data collection

Our engaged fieldwork connected us directly with open source community members and with interactions aimed at understanding the dynamics of open source community health. This connection allowed us to conduct interviews with open source practitioners about how open source community health is defined and understood. All members of the research team were engaged participants (Van de Ven, 2007) and were fully revealed as researchers while contributing to and interacting with corporations and open source communities. Through these engagements,

[2]https://www.linuxfoundation.org/.

we collected data through interviews to explore how social comparison was used by open source community members from a variety of open source communities (Spradley, 2016b) and field notes to capture our own reflective observations and thoughts (Emerson et al., 2011; Spradley, 2016a). Collectively, interviews and field notes constituted the data the research team used to observe, capture, and analyse the shared experiences and perceptions of participants about open source community health and social comparison (Van Maanen, 1988). Specifically, the interviews allowed us to capture the perceptions of open source community members and the field notes allowed us to supplement these perceptions through our own first-hand experience with the language, concepts, practices, rules, and beliefs of community members (Dourish, 2014). The field notes were not analysed using content analysis methods but provided a mechanism through which the research team could collect and reflect on their observations and experiences throughout the field study.

From 2016 to 2023, our research team has primarily been embedded in open source via the CHAOSS[3] project—a Linux Foundation project focused on the definition and deployment of indicators and supporting methods and technologies for understanding open source community health. Figure 1 illustrates the CHAOSS project as a vertical open source community that addresses common issues (e.g., health, security, or licensing) for other horizontal open source communities that focus on precise open source outputs (e.g., kernel or virtualization) (Germonprez et al., 2020).

The CHAOSS project identifies and defines community health metrics and creates software tooling[4] to help community members from different communities understand the health of their projects. Members of our research team are co-founders of the CHAOSS project and currently serve on the CHAOSS board of directors. From our embedded field position, we connected with and listened to community members from across the Linux Foundation open source community ecosystem—as the CHAOSS project touches many corporations and open source communities. Our position within the CHAOSS project enabled connections with interview participants for whom open source community health is a key concern.

We conducted 38 interviews with people engaged in open source communities. The interviews were, on average, 45 min long. The study included interviews with individuals identified as developers, community managers, and

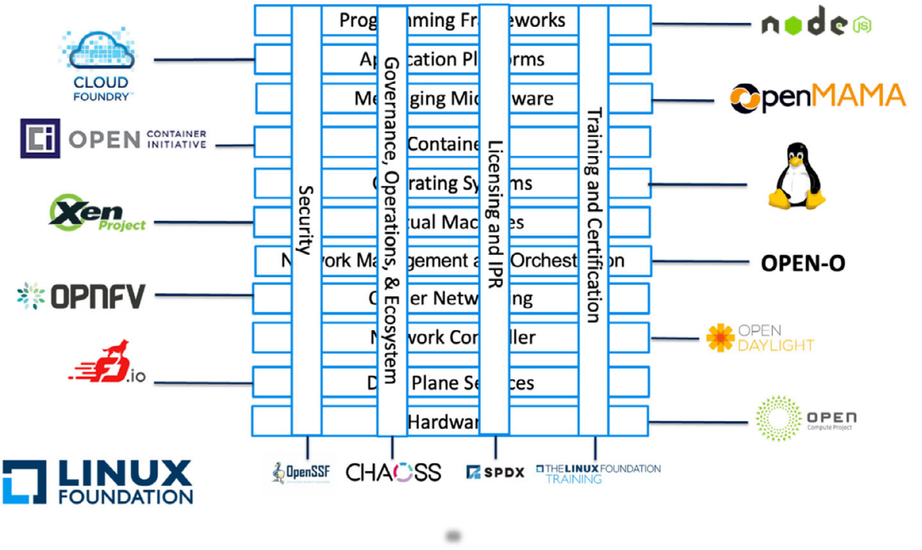

FIGURE 1 "Horizontal" and "Vertical" Open Source Communities as Brokered by the Linux Foundation (Germonprez et al., 2020) and Reprinted with Permission from the Linux Foundation.

[3]https://chaoss.community/.
[4]Augur: https://github.com/chaoss/augur & GrimoireLab: https://github.com/chaoss/grimoirelab.

programme managers who actively contributed to open source communities with significant corporate engagement. Interview participants were identified through our engaged fieldwork and selected based on their interest in open source community health. This interest was identified through participation in the CHAOSS project, attendance at related conferences, and snowball sampling. The participants represented 29 organizations engaged in open source communities, including large organizations such as Red Hat, the Apache Software Foundation, Twitter, Intel, the Linux Foundation, Samsung, and Microsoft. We do not report specific demographic information to ensure we do not jeopardize the anonymity of our informants.

Two interview protocols were used for the study (see Appendix A). The first protocol included 30 interviews and explored context, knowledge, expertise, inspiration, and information gathering (Eckert & Stacey, 2000) in the development of open source community health metrics. This protocol included questions about *why measuring open source community health is important*, *how open source community health is understood across communities*, and *how the effectiveness of open source community health indicators is measured across communities*. During the analysis of these interviews, our research team identified compelling and recurring statements from participants relating to their perceptions about how open source communities use comparison to understand the health of their projects. To explore these ideas in more detail, we created a second interview protocol to investigate the connection between comparison and open source community health more precisely.

The second interview protocol included eight interviews and was created to follow up on very particular aspects of social comparison (Festinger, 1954) as stemming from the first set of interviews. This interview protocol included questions about *how a shared understanding of open source open source community health is created among community members*, *how communities are used as benchmarks*, *how comparison is used to make decisions*, and *how relevant are open source open source community health comparisons*. To ensure that the interview protocols were aligned with the research contributions and appropriately constructed for the research study, we had multiple sessions where we collectively edited and discussed the protocols. Following the completion of the protocols, a pilot interview was conducted for each of the protocols with an expert from outside the research team to ensure that the questions were understood and appropriate for the context.

5.2 | Qualitative content analysis

All interviews were transcribed and analysed using NVivo software,[5] which supports qualitative analysis methods (Mayring, 2014) and the development of themes across interviews (Braun & Clarke, 2019; Creswell & Creswell, 2017; Miles et al., 2014). Through the exploration of literature and our engaged field study, we developed the language to identify and relate social comparison with open source community health. The transcript text was analysed using deductive and inductive content analysis techniques (Mayring, 2014). Our unit of analysis was perceptions about how open source communities use comparison to understand community health. Codes in the interview transcripts were developed iteratively in two ways: first, by deductively mapping theoretical concepts from literature and second, by inductively developing and mapping new concepts to the interview text based on our interpretations of the participant responses. The codes were captured in a handbook in parallel with the analysis. Figure 2. visualizes the handbook construction process, and Table 2 shows key codes and descriptions captured in the handbook.

Interviews were coded in blocks of 3 to 4 sentences, using the coding handbook as a reference. From the 38 interviews, 140 blocks of text were coded (see Appendix B for an example). Through iterative analysis, the coded categories were grouped and organized into themes. Themes are the "analytic outputs developed through and from the creative labour of our coding. They reflect considerable analytic 'work' and are actively created by the researcher at the intersection of data, analytic process, and subjectivity" (Braun & Clarke, 2019, p. 594). The authors repeatedly reread the interviews and the coded quotes to construct themes and identify relationships between concepts.

[5] https://www.qsrinternational.com/nvivo/home.

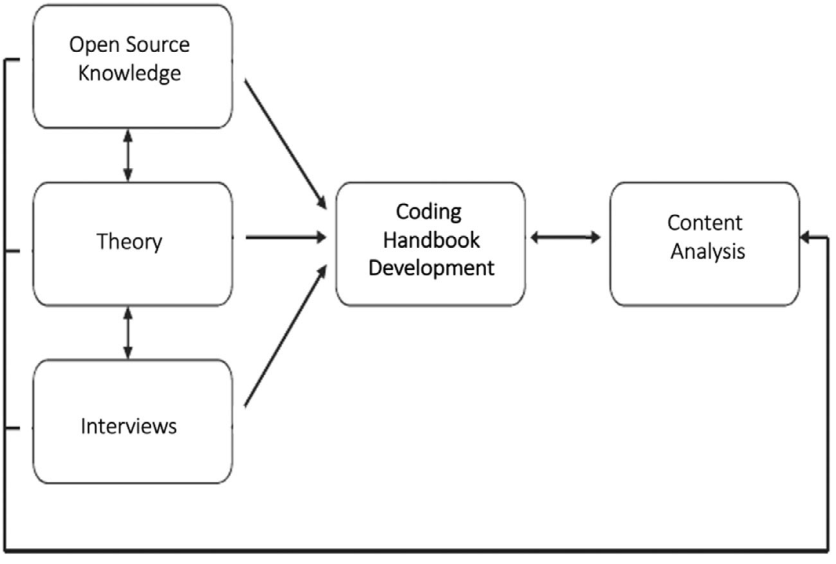

FIGURE 2 Iterative process for Code Handbook development (DeCuir-Gunby et al., 2011).

We constructed themes from the bottom up by organizing codes and concepts into increasingly more abstract categories. This process was iterative and moved back and forth between themes and codes until we had established a set of themes that narratively addressed the research issues (Creswell & Creswell, 2017).

Themes represent broad units of information that consist of insight from the field study, concepts from literature, and codes or several codes aggregated to form explanations that have a high degree of generality (Creswell, 2013; Spradley, 2016b, 2016a). As such, "themes are creative and interpretive stories about the data, produced at the intersection of the researcher's theoretical assumptions, their analytic resources and skill, and the data themselves" (Braun & Clarke, 2019, p. 594). The themes we identified help to explain, for example, why something happened, what something meant, or how the interview participants understood a topic of interest. Our analysis to identify themes was collaborative and reflexive, designed to develop a richer, more nuanced reading of the data (Braun & Clarke, 2019). Appendix B shows the theme development procedure.

Descriptions of the themes were written, as revealed in our Analysis section, which added to our knowledge of how open source community members used social comparison in relation to open source community health. The descriptions came from our reflections on our engaged field study, using a selection of quotes that best expressed the themes developed from our empirical analysis of the interviews. Through our engaged fieldwork, we developed a deeper understanding of open source, building on our reflective experiences to enable a grounded interpretation of social comparison within the context of open source community health (Van Maanen, 1988).

6 | ANALYSIS: SOCIAL COMPARISON AND OPEN SOURCE COMMUNITY HEALTH

Our research investigated how open source community health is understood by participants. We found that open source community health is situated in a social setting where people not only examine discrete indicators but compare those indicators to other communities; they are subjectively determined to be similar in some important way to

TABLE 2 Key codes and descriptions used in the coding handbook.

Code	Description	Supported by literature
Need to evaluate	There is a need to quantify (evaluation) meaningfully and credibly through measurement.	Festinger (1954); Jansen (2014); Nakikj & Mamykina (2018); Naparat et al. (2015)
Comparison relevance	Community comparisons may be (relevant) or (not relevant).	Aué et al. (2016); Festinger (1954); Introne & Drescher (2013); Zgraggen et al. (2018)
Exemplars	Healthy communities may offer best practices. Communities are used as (exemplars) to inform change.	Festinger (1954); Naparat et al. (2015)
Guidance	Social comparison is used to (guide) decisions. Aspects of a project may be changed to improve health.	Arrow et al. (2004); Bietz et al. (2012); Festinger (1954)
Community health	Open source (community health) may equate with success. Communities have a shared definition of "what is good" or health.	Benbya & Belbaly (2011); Crowston et al. (2003); Daniel et al. (2013)
Uniqueness	Communities are (similar) or (different) from one another.	Festinger (1954); Tajfel & Turner (1978)
Ecosystem	Communities may be (standalone) or exist within a collection or (ecosystem) of projects.	Jansen (2014); Jergensen et al. (2011)
Competitive	Open source community members compare communities for (non-competitive) or (competitive) purposes.	Germonprez et al. (2014); Turner (1975)
Standardization	Communities use (standard) or (contextually specific) indicators to understand health.	Festinger (1954); Yuan & Gay (2006)
Narrow focus	There is a tendency to (narrow) or (broaden) the range of comparison.	Festinger (1954)
Modality	An open source community member's position is central (internal) or on the edge (external) of the community.	Blincoe et al. (2012); Festinger (1954); Goggins et al. (2011)

the community they wish to evaluate. Surfacing these types of comparisons, we advance our understanding of open source community engagement as operationalized by acts of social comparison. Our analysis of the data generated three themes that describe how people understand the health of their communities by (1) searching for comparable communities, (2) identifying indicators used by community peers, and (3) using comparison to understand and navigate open source community health. As understood in our qualitative study, the indicators themselves are objective measures of work in open source, and our findings show that the choice of communities and indicators for comparison is often subjective. This demonstrates a process of open source engagement, which appears on the surface to rely on objective indicators but is quite specialized as described next.

6.1 | Searching for comparable communities

People understand the health of their open source community by comparing one community with other communities they identify as similar. When locating communities for comparison, our informants begin by defining a search space of familiar projects. While imperfect, as "all health indicators are a local consideration" [ia_c35],[6] the process of identifying comparable communities is grounded in the experience of the individual performing the comparison. People describe considering the organizational context they are in when selecting communities for comparison. For example, different industries have distinct motivations for engaging in open source, derived from perceptions of market differentiating product features or distinctions in the software development lifecycle a community adheres to [sc_b08; sc_b22]. This begins with an initial coarse-grained assessment and is followed by a narrowing of the field for comparison using specific, subjectively determined indicators that signal adjacency useful for comparison:

> It is basically a simple logical tally. We say we're going to compare our project with another project for this one [metric]. Because that's where we mostly make our comparisons—on [specific] things. So, where are the areas that we are the same as that other project, and where are the areas that we are different
>
> —[sc_b22]

Focusing on the role of technology, some open source communities are more or less technically complex than others [sc_e33]. If two communities are similarly complex along the dimensions framing the comparison, then it is useful [ia_37]. When there is difficulty identifying similar communities, people describe drawing comparisons with another community as "most similar within an imperfect set of choices" [sc_b22]. This is significant because it underscores the complexity of assessing open source community health and how meaning is derived from indicators even when the comparison is imperfect [sc_b08; sc_f32]:

> One of the topics of discussion was: Do those differences [eclipse] the similarities? Is distributing RPM or dev files sufficiently different from distributing container images, or is it just another form of packaging with a different target platform, but the same logical rules of production? I think the general argument was that it was similar enough to look at the same set of issues.
>
> —[sc_b22]

Selection of indicators to use when comparing communities is a heuristic process of identifying the indicators most useful for the intended comparison, which people describe as "close enough." These "close enough" filtering indicators are then applied to generate a list of communities for comparison [sc_k27]. For example, if a community is growing rapidly, a person might search for other rapidly growing communities in the long history of open source software or at present times [sc_a23]. Should a broad assessment of similarities identify other indicators that are useful for comparison, the communities that fit into this emerging dimensionality of indicators and context will be chosen for comparison [sc_b22]. In this way, the selection of both communities for comparison and indicators used in comparison is a reflexive process.

Sometimes, the comparison selection process may be narrowed using more fine-grained indicators. The technical nature of open source community work enables this move to minutiae because it generates a great deal of data, from which indicators are easily derived. Each person develops their *standard indicators* derived from granular data to sift and winnow through candidate comparisons:

> What I'm trying to do is make sure I'm making an apples-to-apples comparison with infrastructure. I'm kind of lucky in the sense that every one of the projects I'm looking at, except one of them, is on GitHub. I look at all the projects, and I look at the pulse indicators and the monthly stats, and I look at the same stats. So, I'm getting honest apples, and that tends to give me enough interesting information.
>
> —[sc_k27]

[6]Inline interview citations denote that this statement is supported by specific interview data. The narrative is built on ideas that were similar across multiple interviews. However, for brevity, we only chose the citation that best exemplified the idea.

As comparisons become active once a community is judged "similar enough for a useful comparison," the act of comparison is likely to involve the exploration of specific instances of communication within a project. Looking at project messaging is a common strategy for making subjective judgements about the similarities of "project culture and practice." How people talk with each other in comments around issues, mailing lists, and discussions about code are signals of project culture used to validate the selection of a comparison.

These project culture-centred comparisons are centred on the visible social aspects of a community [sc_a23]. The nature, location, and frequency of community communication [ia_g03] define project culture. Communities that release software frequently, sometimes called *high-velocity communities*, may feel less personal to newcomers because efficiency may require abrupt communication (i.e., the Linux kernel). Given a project with abrupt communication and high velocity, one might reasonably seek a similar community for comparison. Lower velocity communities are often found within established, longer running, and infrastructural projects. These communities have more and more stable "working code" out in the open and may, therefore, spend more time deliberating a proposed feature change via email lists or issue tracking software [sc_b22] to have more certainty about the impacts of proposed changes. These, and other differences in communication practice, shape each community's reputation in ways that influence another community's assessment of similarity [ia_m25]:

> Understanding the different ways in which the people, the technology, or the user community are similar or more familiar that let them communicate or develop or use things. The main thing is I am usually looking for a sense of what's familiar. For example, does this project feel like one of these projects, and which aspect of it, is it a code, is it the user community, is it the purpose [of the] community? I guess it becomes just a question of community-to community identity.
>
> —[sc_a23]

Sometimes, the absence of similarity in one dimension can motivate a project comparison. This is most often the case when an external community has a reputation for doing "one thing" well (e.g., welcoming newcomers) and is treated as an exemplar to compare with [sc_f32]. When people compare their communities to exemplar communities on the "one thing," a balance between identified indicators of that "one thing" and proxy metrics that are viewed as corresponding with that "one thing" are components of how social comparison is operationalized for a local project [sc_l12]:

> What a Ruby developer expects when they come to a project is different from what a GNU developer expects when they come to a project. That is definitely one way in which I compare projects. I'll try to pick one or two of those projects in that space, as sort of my examples that I'll work from and say, "We're just going to make a little comparison. Are they doing this? How are we doing this?" For some things, it can be really important to learn from those good examples.
>
> —[sc_a23]

In summary, community members search for comparable communities based on perceived community similarities, even if this perception is somewhat imprecise (see Figure 3). Comparisons are often based on specific community similarities but can also occur as community members narrow their focus to compare with communities using a collection of precise indicators that a community is known to be good at.

6.2 | Identifying indicators used by peer communities

Identifying similar communities or exemplar communities for comparison is subjective. Perhaps surprisingly, the quantitative indicators of project health used in any given comparison are subjectively identified as well. Specifically, our informants described selecting these indicators using heuristics, individual experience, and socially constructed

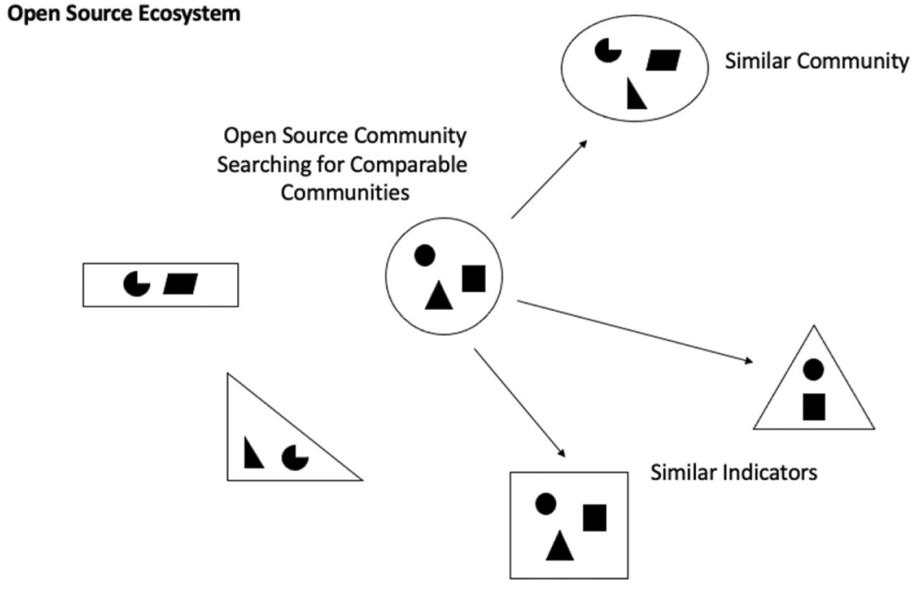

FIGURE 3 Finding similar communities and similar indicators.

influence from others. The result is an idiosyncratic process masquerading in the garb of apparently quantitative, objective indicators.

Each new comparison evolves from prior comparison-indicator combinations and some uptake of prior indicator use within those combinations. Our informants describe how the social comparison influence extends to the identification of specific open source community health indicators applied in a given comparison [ia_a37; ia_h16]. The sociality of this process extends to the uptake and continued use of specific indicators by other projects. Effectively, social comparison undergirds both the identification of comparable communities and the metrics and indicators used to assess health:

> The more people you can talk to who are looking at things from different angles, different end goals, I think that would help come up with health indicators. Then, if you look at some examples of what you think are relatively healthy projects and then see how they actually stand up when you use your indicators, are they coming up as healthy? If they truly are a great project, but your indicators don't make them look good, then you may want to rethink what you're doing.
>
> —[ia_a09]

The use of one indicator by a community influences the likelihood that others will use that indicator in project comparisons [sc_c21]. There becomes, in practice, a general acceptance of project health indicators, which is derived from the influence of other projects, as well as authority in some cases. For example, knowing a government agency uses a set of indicators to deem the airplane you are about to board as airworthy [sc_f32] is similar to the authority that an influential project can exert within an ecosystem. The adoption of health indicators is fully informed by observing how other people use them:

> It is a set of discussions where we are looking into the context and the community's life to see how they are using the indicators. Whether they think it is useful for them or not, whether they think that they are able to balance out their ecosystem better, or whether they can predict some things that affect the health of their communities. Whether or not they think that they understand the dynamics of their own community better.
>
> —[ia_h16]

Social comparisons can lead to a clique-like form of influence, within which the use of a specific health indicator by multiple projects increases the likelihood that more projects will adopt that specific indicator [sc_a23]. This influence can grow if it becomes clear that attention to a particular health indicator is changing a project in ways that are considered desirable. In that case, that indicator is likely to be more widely adopted as central for conducting comparisons:

> I've always liked the way OpenStack has put developers at the center of what they try to do. If you go to their events, their developers are just well-treated, and they've also done a diversity and inclusion report for several years. That's one community that I often look at. And for various reasons, I've kept in touch with people in that community.
>
> —[sc_f32]

> We're just going to have to say, "We've not been able to validate this theory," but it's a theory, and there may be a community for which it is true. As we begin testing more of these theories against communities, we may come to understand the meta complexities that communities are organized around solely code contribution. [Such] communities differ greatly from communities like Fedora, which are not organized primarily around code contributions. The same question is not answerable in both.
>
> —[ia_d06]

Asking another project or peer, “what indicators of project health do you use?” is, in an open source context, akin to asking for advice on balancing an investment portfolio in a financial advisor context. The answer to such a question reveals useful information but must also be localized to the goals of each specific project or person. If a person learns something, they view as insightful for their community, that often triggers wider adoption of a new indicator. That adoption is then considered in the local context of a community.

> Aside from just talking to people and asking “What indicators are you using? Which ones do you find helpful?” Perhaps you want to have some analytics built into what you're doing. For example, we're using the GitHub APIs. You could find which APIs we found are useful just by how often we call them. You get “Hey, these people are really using this a lot.”
>
> —[ia_a09]

Though project health indicators are objective, quantitative signals, interpretation of specific values, and the utility of specific indicators centre on the socially constructed goals of an open source community. As people select indicators to orient themselves and to reflect on their project lifecycle [ia_k13; ia_j14], indicators used by similar communities are localized to highlight community-specific concerns [ia_a37; ia_c07]. Health indicators are most often quantitative, but the interpretation of specific values is usually tuned to the circumstances of a particular open source community.

> There are all these groups, and they are looking at very similar or the same numbers and get to completely different conclusions and ideas on what's happening and what should be done. It is important to understand the points of view and to understand what we are trying to do with the [indicators] ourselves. When we are creating an indicator, we create an indicator which is agnostic, and then how we can help all these different groups, to make a good sense of it, and put a context around it. I don't think that the indicators have good use and meaning without the context where we are using them.
>
> —[ia_h16]

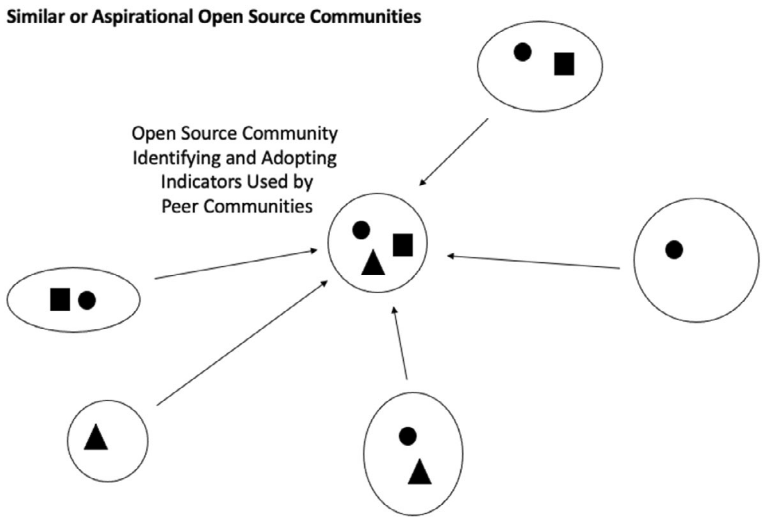

FIGURE 4 Identifying and adopting indicators used by peer communities.

> Our project is quite a bit different. Because of the complexity[…] The kind of health indicators that would make sense for us might be a little different than for some other projects. Every project is different.
>
> —[ia_37]

In summary, identifying indicators is based on their adoption by other communities that are identified as similar as related to project-specific outcomes (see Figure 4). Identification is further based on an indicator's applicability in a local context. While many communities may adopt the same health indicators, the interpretation of health indicators is generally contextualized by domain and software development process factors at the individual community level.

6.3 | Using comparison to understand and navigate open source community health

Navigating the application of discrete indicators and community comparisons that each emerge principally through a process of social comparison demands continuous cycles of orientation, decision-making, and navigation. Social comparison is not static or frozen in time. Instead, our informants describe a process through which it is not always clear which communities or indicators make the optimal comparisons. Making comparisons, therefore, always requires a moment of orientation to situate oneself in a space. When an indicator is identified, navigation towards comparable projects follows; when a comparable project is identified, navigation towards appropriate indicators ensues. Social comparison drives selection, and ultimately it drives navigation or evaluation of a comparable project. Figure 3 provides a representation of the orientation, decision-making, and navigation processes that underlie the development of both communities that are candidates for comparison and the indicators used in those comparisons. The curved lines reflect how three distinct axes are constantly adjusted in practice.

People describe how healthy practices propagate through social comparison of health indicators and a dynamic that develops between knowing project health and getting stronger [ia_b36]. If, through comparison, a person learns that their community runs the *slowest metaphorical mile* among its peers, that information can be applied to improve a project:

> We're interested in benchmarking our community projects to identify weak points or areas for improvement and areas where we are excelling.
>
> —[ia_i01]

Understanding health then becomes a dynamic of (1) finding communities to compare with, (2) identifying indicators, and (3) drawing this information together in the achievement of community-specific health goals. People use comparisons as ways to navigate the complex environment of open source community health for their projects [ia_d20; ia_i01].

> Are we on the right path trends against where we have been and against what's going on with the other parts of the ecosystem? One of the things I was doing is looking across all the [comparable projects] every month and seeing—how many commits are there, how many new developers are there, and if we're trending better on those two indicators than the others. I've been using that as an indicator that we're doing something right.
>
> —[sc_k27]

> Half the discussion was, what did the Linux kernel do, and how did that work out? What did these various Linux distributions do, and how did that work out? In a lot of cases, if we're looking at making any kind of change, particularly to the process or management of our project, the first discussion that we have is, has anyone else done this before? How did it work out? I'll frequently go to one of my areas of contact outside my project, one of the various forums or social media sites, and say, hey, has anybody done this? What happened?
>
> —[sc_b22]

Indicators employed in any specific comparison between communities are expressed as objective indicator values that make the comparisons clear through an interaction between the technical, apparently precise values expressed and the social process of comparing values across communities [ia_d06]. To derive meaning from this intersection of data and heuristics, people build guides from which to manage community health. They employ comparison to, for example, seek insights related to technical development and running community events. In both cases, people explore projects within their field of view, building a repertoire of guides used to make assessments of project health [sc_a23].

Open source community health includes building a structured view by which a community can repeat successful efforts—guides that direct navigation through dynamic open source software ecosystems.

> We have several open source communities and one of them we recently donated to a [foundation] so we're looking at other [Foundation] projects to see how they are doing things to make sure that we're not just falling in line but rather see what they are doing to be successful, to make sure that we can be successful within the [foundation] ecosystem, as well. Then for other projects, yes, absolutely, we look at others to see what works for them. Maybe also talk to maintainers and listen to what does not work for them, so we don't make those mistakes. So yeah, we take a lot of input from other projects when we grow and try to sustain the community around our own.
>
> —[sc_c21]

> [My project] is part of an interlocking ecosystem with a whole bunch of other projects. And one of the things that we must constantly do is check with his other projects about if we do X, how is that going to affect you. We did X, has this helped you?
>
> —[sc_b22]

Social comparison provides an identifiable, specific collection of advantages for people to understand open source community health in context. This includes (1) finding communities to compare with, (2) identifying appropriate indicators, and (3) using the accumulated experience of performing those actions to manage community health. Social comparison provides a central orienting mechanism through which open source communities can understand how people make sense of complex open source engagements.

In summary, open source health indicators help create reference points for open source community comparisons and health-related issues. Knowing the most relevant communities for comparison and having a repertoire of indicators enables navigation towards specific community health goals. Understanding and managing community health is a competency developed through the systematic application of specific community health indicators to a collection of subjectively determined comparable projects to advance a community's goals.

7 | DISCUSSION

We framed our research around understanding how people use open source community health to build a comparative understanding of their own communities. In this section, we first discuss how open source community health is an ongoing construction, including how people consider health in relation to others. Second, we discuss how open source can be understood as a collection of communities and how social comparison provides a vehicle for exploring how different open source communities interact in distinctly local ways. Third, we discuss how social comparison can be framed by social and technical considerations, bounding fields of view within which open source communities interact. Fourth, we discuss a series of conjectures and trials on Festinger's (1954) social comparison theory applied to open source software that suggests new relationships between Festinger's hypotheses and derivations. Reflecting Weick's (1989) argument for middle-range theory development, we make assessments of interest that these conjectures are grounded in fundamental differences with assessments Festinger made in one, technologically limited context. We suggest a middle-range theoretical extension of Festinger's (1954) prior work as a contribution to the wider literature on social comparison, applied in the context of open source work.

7.1 | Open source community health as a social comparison

Prior research has primarily explored the health of software artefacts or individual projects, often independent of the relative relationship between communities. In open source community health research, investigation of the artefact is often a focus. For example, projects can be forked, meaning an open source project may survive the collapse or shift of a community and may even see a new community emerge to support the software (Nyman & Lindman, 2013). Open source communities remain healthy in some form as long as the related software is being copied, used, and developed (Gamalielsson & Lundell, 2014). However, we found that people and organizations engaging in open source also seek assurances that they and their communities are doing well relative to others. Open source health indicators have meaning when their values can be compared and understood across communities. For example, the number of code contributions to a project may inform community health by signalling the size of a community or how much work is being done, but what number constitutes a healthy project? The importance of the number of code contributions to the health of a community can be better understood when communities compare themselves with similar communities, using similar indicators. Our research extends views of open source community health (Crowston et al., 2006; Crowston & Howison, 2011; Daniel & Stewart, 2016; Germonprez et al., 2018; Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021; Link & Germonprez, 2018; Naparat et al., 2015) to show that it is a social construction that includes relationships between people, organizations, and technology.

With the steady rise of corporate engagement, we know that open source communities have become an increasingly important part of product innovation strategies (Germonprez et al., 2020). Further, we know that information asymmetries exist in these complex environments, and these asymmetries reflect boundaries to be overcome. In this context, open source community health indicators can help to “convey product quality information to consumers, reducing uncertainty and facilitating a purchase or exchange” (Wells et al., 2011, p. 373). In proprietary software development projects, objective measures denote success, reducing information asymmetries (Saunders & Brynjolfsson, 2016). Stakeholders know a project is important by looking at market factors such as sales, market share, or customer base (Gallaugher & Wang, 2002). The projects that appear to be ahead of the competition may have the largest market share or the highest sales margins. However, these measures are largely absent from open source communities (Lee et al., 2009), even though those engaged in open source communities are driven by the same need to know not only who the competition is but also where open source communities of interest reside in a larger software landscape (Jansen, 2014). Unlike measures of market value that have shared and discoverable meaning to all, open source health indicators are highly contextualized and often asymmetric due to their sociotechnical nature (Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021).

Through social comparison, “there is a unidirectional drive upward in the case of abilities … oriented toward some point on the ability continuum slightly better than their own performance or the performance of those with whom they are comparing themselves” (Festinger, 1954, pp. 126–127). In communities, people seek to overcome information asymmetries and move towards uniformity, as well as a measurable environment from which a comparative view of open source community health emerges. Social comparison drives an exchange of information on open source community health that creates uniformity, and uniform indicators can lead to rivalry as inequities can now be identified between communities.

When two open source communities provide approximately the same services, health indicators can serve as a way of establishing value between the two communities. For example, when an open source community consumer is making decisions about which of two or more competing projects to implement on their infrastructure, an estimation of whether that project's community is healthy enough to be sustainable is a vital component in decision-making. Within open source, a market may be positively served by competition on health indicators. When selecting communities that are competing in the same space, using health indicators as a measure to understand a community's place against others can prove valuable for the entire ecosystem (Festinger, 1954). In the absence of uniform health indicators and subsequent comparisons, open source communities are not able to be understood in uniform ways, and subsequently, high-performing (and alternatively low-performing) projects cannot be identified. Improving the comparability to others allows projects to indicate their health within competitive spaces (Turner, 1975) and allows for the identification of marketplace leaders (e.g., Kubernetes[7] in the container space) as well as marketplace failures (e.g., OpenSSL[8] in the security space during Heartbleed).

The definition of open source community health has drawn heavily on a project's ability and likelihood to produce quality software (Crowston et al., 2006; Crowston & Howison, 2011; Daniel & Stewart, 2016; Germonprez et al., 2018; Goggins, Germonprez, & Lumbard, 2021; Goggins, Lumbard, & Germonprez, 2021; Link & Germonprez, 2018; Naparat et al., 2015). Yet, through social comparison, we found that open source community health is not just a measure of an ability to produce quality software, but it is also about ways of working and relationships between people and communities. Community health is not just one thing (i.e., a software artefact or a single metric), but it is an ongoing social construction, a construction observable through social comparison.

7.2 | Open source as a collection of communities

Using social comparison to investigate open source communities revealed that members reference other communities in their work. When people compare one open source community with their own, they get a sense of how they

[7]https://kubernetes.io/.
[8]https://www.openssl.org/.

are doing, relatively speaking. A comparison is social insofar as people draw comparisons between communities that are similar in terms of personnel or community characteristics. People relied on social comparison to orient their work subject to their perceptions of such things as community value (Festinger, 1954; Tajfel, 1974; Tajfel & Turner, 1978). Through social comparison, people identified indicators that drew essential contrasts, and through these contrasts, they evaluated, for example, whether socialization success actions or the construction of new roles are likely to support the continued success of their open source community.

Considering open source as a collection of communities is not new. Consistent work has been published that presented open source as a "collection of software products that have some given degree of symbiotic relationships" (Yu et al., 2007, p. 75). However, social comparison enabled us to view open source as a collection of communities absent of symbiotic relationships. While we demonstrate, like prior research, open source communities do not live in isolation and rely on similar communities, we find that the relationship may not be one of mutual benefit. Comparison allows for an external evaluation of similarities and differences to support community efforts on such things as improving the newcomer experience or improving issue resolution time. While this one-way engagement certainly demonstrates that open source communities learn from similar communities in their field of view, it is distinctly different from communities and organizations working together to solve shared technical problems (Kabbedijk & Jansen, 2011; Messerschmitt & Szyperski, 2005; Xiao et al., 2018).

Social comparison helped us demonstrate how open source communities build an understanding of the projects around which they work. Motivated by an array of considerations, many open source projects use related or similar open source communities as points of reference and use comparison as an instrument to improve their understanding of open source community engagement. To overcome community challenges, we found that members often march towards uniform indicators and often draw from unaware partners, for assessing open source community goals. We saw communities bend themselves towards homophily as defined by our informants seeking similarity (Tajfel & Turner, 1978), forging patterns of uniform work, and drawing conclusions built from an increasingly similar cohort (Nahon & Hemsley, 2014; Yuan & Gay, 2006).

Through our research, social comparison helped reinforce that software ecosystems are not only defined by technical characteristics but also social characteristics. While open source is, in part, singular communities working on technical solutions, open source is also the networks of people present across communities (Hu & Zhao, 2008; Mens & Goeminne, 2011; Zhang et al., 2014). In particular, we saw that emergent, widely available, and sophisticated approaches for assessing open source community health enabled our work to deeply consider the context of open source as a collection of communities.

7.3 | Social and technical frames for comparisons

When people made sense of different contexts, they slowly built up an opportunity to understand shared and distinct characteristics of different open source software communities. Open source communities aim for a clear understanding of open source community health in the social and technical environments where the complexity of the environment is significantly greater than any one project (Simon, 1996). Yet, a path towards open source community health as a lever for seeing the relationships between collections of open source communities was sometimes bound by the open source projects in a community's field of view. "When a discrepancy exists with respect to opinions or abilities, there will be tendencies to change one's own position so as to move closer to the others in the group" (Festinger, 1954, p. 126).

Open source community health can be bound by the decisions people make in complex situations, shaped by present social and technical structures. In open source, some communities in a field of view are organized in social structures that provide projects with needed communications and organizational support. Brokers like the Linux Foundation or the Eclipse Foundation may frame the field of view within which open source communities find projects to compare themselves with (Fitzgerald, 2006; Germonprez et al., 2017). In doing so, community members self-select into Foundation brokered projects that share similar members, governance structures, and event schedules (Germonprez et al., 2018, 2020; Shaikh & Henfridsson, 2017). Working in proximity to similarly brokered open source communities can frame health comparisons to include open source communities that fit an overall trajectory of their brokering organizations, like the Linux Foundation. This framed space creates pressure towards uniformity in health comparisons (Festinger, 1954). As such, the Linux Foundation, as an organizing structure, reduces the complexity that is the entirety of all open source communities by narrowing the open source community landscape to those with significant corporate engagement but also may frame how social comparisons are identified and carried out.

A field of view can also be framed through the infrastructure used by open source communities. For many open source communities, GitHub serves as a primary platform for hosting code developed activities (Cosentino et al., 2016). Open source communities using GitHub are inherently similar in how work is done and tracked over time through such GitHub-specific processes around issue management, pull requests, commits, and code additions/deletions. For communities using GitHub, health indicators are observable through GitHub's online dashboard and the API. These available indicators create uniformity by framing what open source community members can observe and compare regarding community health. Creating consistent work patterns through platforms like GitHub affects what open source community health indicators are considered, tracked, and compared against.

In this context, social and technical frames reinforce open source software as a collection of communities. Comparisons become immediately easier for projects that are structured and work similarly (i.e., issues and pull requests on GitHub) than for those that are structured and work differently (i.e., project-specific combinations of bug trackers and code reviews)—further framing open source community comparisons (Cosentino et al., 2017). The narrowing of focus allows comparisons to be made more easily despite the seemingly disparate nature of projects. As such, open source community health comparisons may not provide a measure of all available health characteristics, but a measure of health characteristics that are available in the social context that community members self-select into and are available in the technical context of platforms for work.

7.4 | Contributing to Festinger's theory of social comparison in a sociotechnical space

Our last theoretical contribution rests on extending Festinger's theory of social comparison (1954) to describe open source communities. An ontological choice to describe open source software communities as a singular phenomenon embeds, perhaps hides, an epistemological assumption that these communities are all similar in critical ways. We think it likely that different communities of open source software are also dissimilar in critical ways. We conjecture that social comparison is complex in a socio-technical work context comprising collections of communities in open source software and requires an extension of Festinger's theory of social comparison (1954). Such an extension is necessary to address the social and non-social levers of how people measure themselves, cultivate their status, seek to change their abilities, and understand the prospects for the communities they are already part of through social comparison overwhelmingly mediated by technology.

Here, we propose a socio-technical extension of Festinger's theory that considers how comparisons derived from technology, text, and organizational-communal work are (a) materially distinguishable from the empirical data used to develop the original theory, (b) not accounted for fully in Festinger's theory, because the information used for social comparison by our informants emerges from more precise, visible, and overwhelming; but also ephemeral information, and (c) engaged in ongoing routines of social comparison, including assessment of individual, and group abilities, as well as the status and hierarchy of projects, communities, and individuals. Our theoretical extension, therefore, considers the mutability of status in open source communities and other socio-technical contexts.

Festinger's (1954) hypothesis V states: "There are non-social restraints which make it difficult or impossible to change one's ability." Breaking down this hypothesis, Festinger elaborates on a distinction between the processes associated with changing opinions and changing abilities in a social comparison. Changing opinions is described as a largely social activity, while changing abilities is said to be restrained by limitations in the environment, and the likelihood of a person changing their abilities, which is considerably less likely than changing opinions. This distinction, specifically, is not easily worked out in the development of open source software. GitHub, which hosts an overwhelming majority of open source software projects today, is inherently a "social coding platform" (Dabbish et al., 2012; McDonald & Goggins, 2013). Work activity on GitHub and other platforms including GitLab and Bitbucket weave work and talk (Goggins et al., 2013) together in a way that significantly mutes the distinction between opinions and abilities.

Our findings illustrate the inherent backgrounding of distinctions between opinion and ability, especially in the social comparison of open source projects. The notion of ability is triangulated through a need within a community to evaluate itself, and subsequent comparisons with other communities identified as generally relevant, exemplars of good practice, placement of a project within an ecosystem, the presence or absence of competition between compared projects, and various emerging standards for measurement of open source community health. Open source communities, unlike groups in a confined, co-located context typically operationalize some form of distributed leadership (McDonald et al., 2014), where maintainers emerge and recede over time. Distributed leadership in open source software is enacted using issue trackers, review of proposed changes that anyone is free to make, and decisions about the technical arc of the project that are sometimes carefully governed, and other times less carefully governed. Influence of individuals on community decisions is not only a function of ability but also well understood to be a function of the amount of social capital assigned to developers through platform stars and following behaviour; more popular developers, a social measure, have more influence on project direction (Blincoe et al., 2016).

Social comparison theory supports people's use of open source community health indicators to understand open source as a construction beyond any single community. The introduction of indicators for project health enables projects to make comparisons with other projects to advance their health in a more systematic way than platform measures, and here, too, we see a blurred line between ability (project health) and opinion. How healthy a project is or is not perceived to be in a comparison is a function of the indicators selected and the internal, community-specific goals driving the comparison. Health indicators in open source software are a moving target for the community making the comparison and the communities being compared. Any disruption to a project ecosystem brings with it a risk of community health, our proxy for Festinger's "ability," rising or declining in relatively short periods of time. Most published open source community health indicators do not make a clear, systematic distinction between changes arising from opinions and changes arising from ability because they are difficult to separate in a technologically mediated collaboration on a social coding platform. Today's comparisons become yesterday's comparisons rapidly.

Festinger et al. (1954) advanced nine concepts that inform the theory of social comparison, centred on a set of assumptions that social comparison is constrained entirely to observable behaviour and communication within a defined group. In the technologically mediated context of open source software, we suggest two new concepts.

1. In cases where work is decentralized and technologically mediated, the socio-technical practices of a community constrain and enable the likelihood that a collection of opinions and abilities results in a material change to the work artefacts.
2. In cases where work is decentralized and technologically mediated, the influence of comparisons with another project on a change in practices to improve community health is the result of consensus, or distributed leadership, within a project.

Underlying these two concepts is our observation that social comparison is a socially networked and technically networked process in open source communities. Each of these networks is fluid, and the process of social comparison within and across communities contributes to this structural fluidity. We therefore suggest that extensions of Festinger's (1954) theory of social comparison applied to open source communities, and we suspect other similarly networked social settings, recognize how the process of social comparison opens up a possibility centred view that enables growth, innovation, and the likelihood of individual and collective abilities and opinions evolving with fewer constraints.

8 | CONCLUSIONS

In contemporary open source engagements, work must be understandable to people, organizations, and foundations in ways that provide clear insights that are recognizable and actionable. Open source community health is *one way* to make work understandable that can be employed to contribute to a broader understanding of open source engagement. Other practices aimed at improving our understanding of open source engagement certainly exist and include, for example, those focused on licence compliance (Gandhi et al., 2018) and software development (von Krogh et al., 2012). Through social comparison, open source community health is more than just a point-in-time observation but a way of better knowing and participating in complex open source environments.

8.1 | Limitations of the current work

While our research serves as an extension of how we consider open source as a complex socio-technical construction, we did not explore how social comparisons impact the ongoing success or failure of open source communities. For example, social navigation (Dieberger et al., 2000) may serve as a platform to understand how open source community health manifests in open source communities as a wayfinding strategy (Mollerup, 2009). Such action-oriented research can have implications as open source communities adopt or possibly abandon shared understandings of open source community health indicators, as well as the dynamics of how socio-technical systems play a role in further bounding social comparison. Additionally, it is unclear from our research how the modality of community members affects the bounding of comparisons. In this, research can explore bounded comparisons further to identify how different types of community members use comparisons (e.g., developers, managers, lawyers, brokers).

People want to know that their projects will continue to serve the needs of the community (Nyman & Lindman, 2013). Thus, questions about sustainability are important considerations of open source community health. Yet, definitions of sustainability are sometimes unclear and often vary based on the context (Bietz et al., 2012; Chengalur-Smith et al., 2010; Connelly et al., 2011; Davison, 2001; McManus, 1996). To sustain open source communities means to make them endure for long and even indefinite periods of time. However, what level of activity constitutes sustainability, and how productive does a project need to be? Numerous approaches can be used to describe and measure sustainability (Bietz et al., 2012; Connelly, 2007; Davison, 2001; McManus, 1996), and open source community health offers one venue to explore what constitutes sustainability (Bietz et al., 2012; Chengalur-Smith et al., 2010; Connelly, 2007; Davison, 2001; McManus, 1996). However, our research is not able to provide an empirical contribution to definitions of sustainability as we did not explore the lifecycle of open source projects in detail. While we know that sustainability is a desirable characteristic for community members, it is not something we precisely observed over time. We posit that community health measures the characteristics of a project that may make it sustainable, but the relationship between open source community health and sustainability needs to be explored in more detail.

Finally, our research focuses on a segment of open source work with high levels of corporate and organizational engagement. While engagement with open source communities is critically important for nearly any technology-enabled organization, this context does not represent all open source communities. For example, many open source communities are not part of an open source foundation or may not need to monitor the health of their open source community. Many open source communities are "open source in licence only" or may have created stable open source artefacts that only require minor security fixes on an irregular basis. In this research, we show that social comparison is a present component in certain open source community contexts and describe its characteristics, but those characteristics may not be fully applicable in every corner of the vast open source ecosystem. More work remains to build a wider view, but contextualization and repositioning of our research are likely needed to help specify how comparison is understood across the complex setting of open source.

FUNDING INFORMATION

Alfred P. Sloan Foundation (grants: https://sloan.org/grant-detail/8434, https://sloan.org/grant-detail/9549, and https://sloan.org/grant-detail/10384).

DATA AVAILABILITY STATEMENT

The data that supports the findings of this study are not publicly available due to privacy or ethical restrictions.

ORCID

Kevin Lumbard https://orcid.org/0000-0001-9306-3040

REFERENCES

Abunadi, I., & Alenezi, M. (2015). Towards cross project vulnerability prediction in open source web applications. *Proceedings of the the International Conference on Engineering & MIS*, *2015*, 42:1–42:5.

Aksulu, A., & Wade, M. (2010). A comprehensive review and synthesis of open source research. *Journal of the Association for Information Systems*, *11*(11), 576–656.

Arantes, F. L., & Freire, F. M. P. (2011). Aspects of an open source software sustainable life cycle. *7th IFIP International Conference on Open Source Systems*, Proceedings 7, 325–329.

Arrow, H., McGrath, J. E., & Berdahl, J. L. (2000). *Small groups as complex systems: Formation, coordination, development, and adaptation*. Sage Publications.

Arrow, H., Poole, M. S., Henry, K. B., Wheelan, S., & Moreland, R. (2004). Time, change, and development the temporal perspective on groups. *Small Group Research*, *35*(1), 73–105.

Aué, J., Haisma, M., Tómasdóttir, K. F., & Bacchelli, A. (2016). Social diversity and growth levels of open source software projects on GitHub. *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 41:1–41:6.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191–215.

Benbya, H., & Belbaly, N. (2011). *Successful OSS project design and implementation: Requirements, tools, social designs and reward structures*. Ashgate Pub.

Bhadauria, V. S., Mahapatra, R., & Manzar, R. (2009). Factors influencing adoption of open source software-an exploratory study.

Bietz, M. J., Ferro, T., & Lee, C. P. (2012). Sustaining the development of cyberinfrastructure: An organization adapting to change. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 901–910.

Blincoe, K., Sheoran, J., Goggins, S., Petakovic, E., & Damian, D. (2016). Understanding the popular users: Following, affiliation influence and leadership on GitHub. *Information and Software Technology*, *70*, 30–39.

Blincoe, K., Valetto, G., & Goggins, S. (2012). Proximity: A measure to quantify the need for developers' coordination. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 1351–1360.

Braun, V., & Clarke, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, *11*(4), 589–597.

Buckingham, J. T., & Alicke, M. D. (2002). The influence of individual versus aggregate social comparison and the presence of others on self-evaluations. *Journal of Personality and Social Psychology*, *83*(5), 1117–1130.

Capiluppi, A., Serebrenik, A., & Youssef, A. (2012). Developing an H-index for OSS developers. *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories*, 251–254.

Chengalur-Smith, I., Sidorova, A., & Daniel, S. (2010). Sustainability of free/libre open source projects: A longitudinal study. *Journal of the Association for Information Systems*, *11*(11), 657–683.

Chengalur-Smith, S., & Sidorova, A. (2003). Survival of open-source projects: A population ecology perspective. In *the proceedings of the 24th international conference on information systems* (p. 66). ICIS.

Chou, S.-W., & He, M.-Y. (2011). The factors that affect the performance of open source software development–the perspective of social capital and expertise integration. *Information Systems Journal*, *21*(2), 195–219.

Connelly, B. L., Certo, S. T., Ireland, R. D., & Reutzel, C. R. (2011). Signaling theory: A review and assessment. *Journal of Management*, *37*(1), 39–67.

Connelly, S. (2007). Mapping sustainable development as a contested concept. *Local Environment*, *12*(3), 259–278.

Cosentino, V., Izquierdo, J. L. C., & Cabot, J. (2017). A systematic mapping study of software development with GitHub. *IEEE Access*, *5*, 7173–7192.
Cosentino, V., Luis, J., & Cabot, J. (2016). Findings from github: Methods, datasets and limitations. *Proceedings of the 13th International Conference on Mining Software Repositories*, 137–141.
Creswell, J. W. (2013). *Qualitative inquiry and research design: Choosing among five approaches* (3rd ed.). SAGE Publications.
Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage Publications.
Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. In *the proceedings of the 24th international conference on information systems* (p. 28). ICIS.
Crowston, K., & Howison, J. (2006). Assessing the health of open source communities. *Computer*, *39*(5), 89–91.
Crowston, K., & Howison, J. (2011). FLOSS project effectiveness measures. In H. Benbya & N. Belbaly (Eds.), *Successful OSS project design and implementation: Requirements, tools, social designs and reward structures* (pp. 149–167). Ashgate Publishing.
Crowston, K., Howison, J., & Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice*, *11*(2), 123–148.
Crowston, K., & Scozzi, B. (2008). Bug fixing practices within free/libre open source software development teams. *Journal of Database Management*, *19*(2), 1–30.
Cummings, J. N., & Cross, R. (2003). Structural properties of work groups and their consequences for performance. *Social Networks*, *25*(3), 197–210.
Curto-Millet, D., & Shaikh, M. (2017). The emergence of openness in open-source projects: The case of openEHR. *Journal of Information Technology*, *32*(4), 361–379.
Dabbish, L., Farzan, R., Kraut, R., & Postmes, T. (2012). Fresh faces in the crowd: Turnover, identity, and commitment in online groups. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 245–248.
Daniel, S., Agarwal, R., & Stewart, K. J. (2013). The effects of diversity in global, distributed collectives: A study of open source project success. *Information Systems Research*, *24*(2), 312–333.
Daniel, S., & Stewart, K. (2016). Open source project success: Resource access, flow, and integration. *The Journal of Strategic Information Systems*, *25*(3), 159–176.
Davies, J., German, D. M., Godfrey, M. W., & Hindle, A. (2011). Software bertillonage: Finding the provenance of an entity. *Proceedings of the 8th Working Conference on Mining Software Repositories*, 183–192.
Davison, A. (2001). *Technology and the contested meanings of sustainability*. SUNY Press.
DeCuir-Gunby, J. T., Marshall, P. L., & McCulloch, A. W. (2011). Developing and using a codebook for the analysis of interview data: An example from a professional development research project. *Field Methods*, *23*(2), 136–155.
Dieberger, A., Dourish, P., Höök, K., Resnick, P., & Wexelblat, A. (2000). Social navigation: Techniques for building more usable systems. *Interactions*, *7*(6), 36–45.
Dourish, P. (2014). Reading and interpreting ethnography. In J. Olson & W. Kellogg (Eds.), *Ways of knowing in HCI* (pp. 1–23). Springer.
Dreyer, A. S. (1954). Aspiration behavior as influenced by expectation and group comparison. *Human Relations*, *7*, 175–190.
Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M., & Halderman, J. A. (2014). The matter of Heartbleed. *Proceedings of the 2014 Conference on Internet Measurement Conference*, 475–488.
Eckert, C., & Stacey, M. (2000). Sources of inspiration: A language of design. *Design Studies*, *21*(5), 523–538.
Eghbal, N. (2016). *Roads and bridges: The unseen labor behind our digital infrastructure*. Ford Foundation. https://www.fordfoundation.org/about/library/reports-and-studies/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/
Emerson, R. M., Fretz, R. I., & Shaw, L. L. (2011). *Writing ethnographic fieldnotes*. University of Chicago Press.
Faraj, S., Kudaravalli, S., & Wasko, M. (2015). Leading collaboration in online communities. *MIS Quarterly*, *39*(2), 393–412.
Feller, J., Finnegan, P., Fitzgerald, B., & Hayes, J. (2008). From peer production to productization: A study of socially enabled business exchanges in open source service networks. *Information Systems Research*, *19*(4), 475–493.
Feller, J., & Fitzgerald, B. (2000). A framework analysis of the open source software development paradigm. In *the proceedings of the 21st international conference on information systems* (p. 7). ICIS.
Feller, J., & Fitzgerald, B. (2003). Is open source revolutionising the software industry?
Festinger, L. (1950). Informal social communication. *Psychological Review*, *57*, 271–282.
Festinger, L. (1954). A theory of social comparison processes. *Human Relations*, *7*(2), 117–140.
Festinger, L., Torrey, J., & Willerman, B. (1954). Self-evaluation as a function of attraction to the group. *Human Relations*, *7*, 161–174.
Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly*, *30*, 587–598.
Foyzur, R., & Premkumar, D. (2013). How, and why, process metrics are better. In *proceedings of the 2013 international conference on software engineering (ICSE '13)* (pp. 432–441). IEEE Press.

Franco-Bedoya, O., Ameller, D., Costal, D., & Franch, X. (2017). Open source software ecosystems: A systematic mapping. *Information and Software Technology*, *91*, 160–185.

Gacek, C., & Arief, B. (2004). The many meanings of open source. *IEEE Software*, *21*(1), 34–40.

Gallaugher, J. M., & Wang, Y.-M. (2002). Understanding network effects in software markets: Evidence from web server pricing. *MIS Quarterly*, *26*, 303–327.

Gamalielsson, J., & Lundell, B. (2014). Sustainability of open source software communities beyond a fork: How and why has the libreoffice project evolved? *Journal of Systems and Software*, *89*, 128–145.

Gandhi, R., Germonprez, M., & Link, G. J. P. (2018). Open data standards for open source software risk management routines: An examination of SPDX. *Proceedings of ACM GROUP '18*, 219–229.

Geiger, R. S., & Ribes, D. (2011). Trace ethnography: Following coordination through documentary practices. In *proceedings of the 2011 44th Hawaii international conference on system sciences (HICSS '11)* (pp. 1–10). IEEE Computer Society.

Germonprez, M., Kendall, J. E., Kendall, K. E., Mathiassen, L., Young, B., & Warner, B. (2017). A theory of responsive design: A field study of corporate engagement with open source communities. *Information Systems Research*, *28*(1), 64–83.

Germonprez, M., Kendall, J. E., Kendall, K. E., & Young, B. (2014). Collectivism, creativity, competition, and control in open source software development: Reflections on the emergent governance of the SPDX® working group. *International Journal of Information Systems and Management*, *1*(1–2), 125–145.

Germonprez, M., Levy, M., Kendall, J. E., & Kendall, K. E. (2020). Tapestries of innovation: Structures of contemporary open source project engagements. *Journal of the Association for Information Systems*, *21*(3), 5–663.

Germonprez, M., Link, G. J. P., Lumbard, K., & Goggins, S. (2018). Eight observations and 24 research questions about open source projects: Illuminating new realities. *Proceedings of the ACM on Human-Computer Interaction*, *2*, 57:1–57:22.

Germonprez, M., Lipps, J., & Goggins, S. (2019). The rising tide: Open source's steady transformation. *First Monday*, *24*(8).

Gersick, C. J. (1988). Time and transition in work teams: Toward a new model of group development. *Academy of Management Journal*, *31*(1), 9–41.

Goggins, S., Germonprez, M., & Lumbard, K. (2021). Making open source project health transparent. *Computer*, *54*(8), 104–111.

Goggins, S., Laffey, J., & Gallagher, M. (2011). Completely online group formation and development: Small groups as socio-technical systems. *Information Technology & People*, *24*(2), 104–133.

Goggins, S., Lumbard, K., & Germonprez, M. (2021). Open source community health: Analytical metrics and their corresponding narratives. *2021 IEEE/ACM 4th International Workshop on Software Health in Projects, Ecosystems and Communities (SoHeal)*, 25–33.

Goggins, S., Mascaro, C., & Valetto, G. (2013). Group informatics: A methodological approach and ontology for sociotechnical group research. *Journal of the American Society for Information Science and Technology*, *64*(3), 516–539.

Gustavsson, T. (2020). Managing the open source dependency. *Computer*, *53*(2), 83–87.

Hahn, J., Moon, J. Y., & Zhang, C. (2008). Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Information Systems Research*, *19*(3), 369–391.

Han, K., Oh, W., Im, K. S., Chang, R. M., Oh, H., & Pinsonneault, A. (2012). Value cocreation and wealth spillover in open innovation alliances. *MIS Quarterly*, *36*, 291–315.

Hannemann, A., Liiva, K., & Klamma, R. (2014). Navigation support in evolving open-source communities by a web-based dashboard. In L. Corral, A. Sillitti, G. Succi, J. Vlasenko, & A. I. Wasserman (Eds.), *Open source software: Mobile open source technologies* (pp. 11–20). Springer.

Hars, A., & Ou, S. (2000). Why is open source software viable? A study of intrinsic motivation, personal needs and future returns. In *the proceedings of the 5th Americas conference on information systems* (p. 379). AMCIS.

Heikinheimo, H., & Kuusisto, T. (2004). The use of embedded open source software in commercial products. In *the proceedings of the 12th European conference on information systems* (p. 65). ECIS.

Hoffman, P. J., Festinger, L., & Lawrence, D. H. (1954). Tendencies toward group comparability in competitive bargaining. *Human Relations*, *7*, 141–159.

Howison, J., & Crowston, K. (2014). Collaboration through open superposition: A theory of the open source way. *MIS Quarterly*, *38*(1), 29–50.

Hu, D., & Zhao, J. L. (2008). A comparison of evaluation networks and collaboration networks in open source software communities.

Introne, J. E., & Drescher, M. (2013). Analyzing the flow of knowledge in computer mediated teams. *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, 341–356.

Izquierdo-Cortazar, D., Sekitoleko, N., Gonzalez-Barahona, J. M., & Kurth, L. (2017). Using metrics to track code review performance. In *proceedings of the 21st international conference on evaluation and assessment in software engineering (EASE '17)* (pp. 214–223). Association for Computing Machinery.

Jansen, S. (2014). Measuring the health of open source software ecosystems: Beyond the scope of project health. *Information and Software Technology*, *56*(11), 1508–1519.

Jergensen, C., Sarma, A., & Wagstrom, P. (2011). The onion patch: Migration in open source ecosystems. *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, 70–80.

Jiang, Y., Adams, B., Khomh, F., & German, D. M. (2014). Tracing back the history of commits in low-tech reviewing environments: A case study of the linux kernel. *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 51:1–51:10.

Johari, K., & Kaur, A. (2011). Effect of software evolution on software metrics: An open source case study. *SIGSOFT Software Engineering Notes*, *36*(5), 1–8.

Johari, K., & Kaur, A. (2012). Validation of object oriented metrics using open source software system: An empirical study. *SIGSOFT Software Engineering Notes*, *37*(1), 1–4.

Kabbedijk, J., & Jansen, S. (2011). Steering insight: An exploration of the ruby software ecosystem. In B. Regnell, I. van de Weerd, & O. De Troyer (Eds.), *Software business* (pp. 44–55). Springer.

Kaur, I., & Kaur, A. (2012). Empirical study of software quality estimation. In *Proceedings of the second international conference on computational science, engineering and information technology* (pp. 694–700).

Kozinets, R. V. (2015). *Netnography*. Sage.

Krishnamurthy, S. (2002). Cave or community? An empirical examination of 100 mature open source projects. *First Monday*, *7*(6).

Lee, S.-Y. T., Kim, H.-W., & Gupta, S. (2009). Measuring open source software success. *Omega*, *37*(2), 426–438.

Lech, M., & Marcin, K. (2017). Continuous defect prediction: The idea and a related dataset. In *proceedings of the 14th international conference on mining software repositories (MSR '17)* (pp. 515–518). IEEE Press.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The Journal of Industrial Economics*, *50*(2), 197–234.

Li, H., Zhang, C., & Kettinger, W. J. (2022). Digital platform ecosystem dynamics: The roles of product scope, innovation, and collaborative network centrality. *MIS Quarterly*, *46*(2), 739–770.

Lindberg, A., Berente, N., Gaskin, J., & Lyytinen, K. (2016). Coordinating interdependencies in online communities: A study of an open source software project. *Information Systems Research*, *27*(4), 751–772.

Link, G. J. P., & Germonprez, M. (2018). Assessing open source project health. In *the proceedings of the 24th Americas conference on information systems* (p. 5). AMCIS.

Liu, M., Hull, C. E., & Hung, Y.-T. C. (2017). Starting open source collaborative innovation: The antecedents of network formation in community source. *Information Systems Journal*, *27*(5), 643–670.

Ljungberg, J. (2000). Open source movements as a model for organising. *European Journal of Information Systems*, *9*(4), 208–216.

Lucia Kim, S., & Teo, T. S. (2013). Lessons for software development ecosystems: South Korea's e-government open source initiative. *MIS Quarterly Executive*, *12*(2), 93–108.

Lundell, B., Lings, B., & Syberfeldt, A. (2008). Open source software in complex domains: Current perceptions in the embedded systems area. *AMCIS 2008 Proceedings*, 42.

Machado, F. S., Raghu, T. S., Sainam, P., & Sinha, R. (2017). Software piracy in the presence of open source alternatives. *Journal of the Association for Information Systems*, *18*(1), 3–21.

Mayring, P. (2014). *Qualitative content analysis: Theoretical foundation, basic procedures and software solution*. Klagenfurt.

McDonald, N., Blincoe, K., Petakovic, E. V. A., & Goggins, S. (2014). Modeling distributed collaboration on Github. *Advances in Complex Systems*, *17*(7n08), 1450024.

McDonald, N., & Goggins, S. (2013). Performance and participation in open source software on GitHub. *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 139–144.

McGrath, J. E. (1984). *Groups: Interaction and performance* (Vol. 14). Prentice-Hall.

McGrath, J. E. (1991). Time, interaction, and performance (TIP) a theory of groups. *Small Group Research*, *22*(2), 147–174.

McGrath, J. E. (2000). The study of groups: Past, present, and future. *Personality and Social Psychology Review*, *4*(1), 95–105.

McManus, P. (1996). Contested terrains: Politics, stories and discourses of sustainability. *Environmental Politics*, *5*(1), 48–73.

Mehra, A., Dewan, R., & Freimer, M. (2011). Firms as incubators of open-source software. *Information Systems Research*, *22*(1), 22–38.

Mehra, A., & Mookerjee, V. (2012). Human capital development for programmers using open source software. *MIS Quarterly*, *36*, 107–122.

Meneely, A., Tejeda, A. C. R., Spates, B., Trudeau, S., Neuberger, D., Whitlock, K., Ketant, C., & Davis, K. (2014). An empirical investigation of socio-technical code review metrics and security vulnerabilities. *Proceedings of the 6th International Workshop on Social Software Engineering*, 37–44.

Mens, T., & Goeminne, M. (2011). Analysing the evolution of social aspects of open source software ecosystems. 1–14.

Messerschmitt, D. G., & Szyperski, C. (2005). *Software ecosystem: Understanding an indispensable technology and industry* (p. 1). MIT Press Books.

Miles, M. B., Huberman, A. M., & Saldana, J. (2014). *Qualitative data analysis*. Sage.

Mollerup, P. (2009). Wayshowing in hospital. *Australasian Medical Journal*, *10*, 112.

Mondal, M., Roy, C. K., & Schneider, K. A. (2012). Connectivity of co-changed method groups: A case study on open source systems. *Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research*, 205–219.

Muller, J. (2019). The tyranny of metrics. In *The tyranny of metrics*. Princeton University Press.

Nahon, K., & Hemsley, J. (2014). Homophily in the guise of cross-linking: Political blogs and content. *American Behavioral Scientist*, *58*(10), 1294–1313.

Nakikj, D., & Mamykina, L. (2018). *Lost in migration: Information management and community building in an online health community*. 1–14.

Naparat, D., Finnegan, P., & Cahalane, M. (2015). Healthy community and healthy commons: 'Opensourcing' as a sustainable model of software production. *Australasian Journal of Information Systems*, *19*, 1–12.

Nelson, M., Sen, R., & Subramaniam, C. (2006). Understanding open source software: A research classification framework. *Communications of the Association for Information Systems*, *17*, 266–287.

Nyman, L., & Lindman, J. (2013). Code forking, governance, and sustainability in open source software. *Technology Innovation Management Review*, *3*(1), 7–12.

O'Mahony, S. (2007). The governance of open source initiatives: What does it mean to be community managed? *Journal of Management & Governance*, *11*, 139–150.

Ortu, M., Destefanis, G., Kassab, M., & Marchesi, M. (2015). Measuring and understanding the effectiveness of JIRA developers communities. *Proceedings of the Sixth International Workshop on Emerging Trends in Software Metrics*, 3–10.

Plakidas, K., Stevanetic, S., Schall, D., Ionescu, T. B., & Zdun, U. (2016). How do software ecosystems evolve? A quantitative assessment of the R ecosystem. *Proceedings of the 20th International Systems and Software Product Line Conference*, 89–98.

Raja, U., & Tretter, M. J. (2012). Defining and evaluating a measure of open source project survivability. *IEEE Transactions on Software Engineering*, *38*(1), 163–174.

Ren, Y., Kraut, R., & Kiesler, S. (2007). Applying common identity and bond theory to design of online communities. *Organization Studies*, *28*(3), 377–408.

Riehle, D. (2009). The commercial open source business model. *Value Creation in E-Business Management: 15th Americas Conference on Information Systems, AMCIS 2009, SIGeBIZ Track, San Francisco, CA, USA, August 6-9, 2009. Selected Papers*, 18–30.

Rosen, C., Grawi, B., & Shihab, E. (2015). Commit guru: Analytics and risk prediction of software commits. *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, 966–969.

Rozenberg, D., Beschastnikh, I., Kosmale, F., Poser, V., Becker, H., Palyart, M., & Murphy, G. C. (2016). Comparing repositories visually with repograms. *Proceedings of the 13th International Conference on Mining Software Repositories*, 109–120.

Saunders, A., & Brynjolfsson, E. (2016). Valuing information technology related intangible assets. *MIS Quarterly*, *40*(1), 83–110.

Sawyer, S. (2000). A social analysis of software development teams: Three models and their differences.

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. (2006). Understanding free/open source software development processes. *Software Process: Improvement and Practice*, *11*(2), 95–105.

Schweik, C. M., & English, R. C. (2012). *Internet success: A study of open-source software commons*. MIT Press.

Shaikh, M., & Cornford, T. (2012). Navigating open source adoption in the public sector.

Shaikh, M., & Henfridsson, O. (2017). Governing open source software through coordination processes. *Information and Organization*, *27*(2), 116–135.

Sharma, M., Kumari, M., & Singh, V. B. (2015). Post release versions based code change quality metrics. *Proceedings of the Third International Symposium on Women in Computing and Informatics*, 235–243.

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). MIT Press.

Singh, P. V., & Phelps, C. (2013). Networks, social influence, and the choice among competing innovations: Insights from open source software licenses. *Information Systems Research*, *24*(3), 539–560.

Singh, P. V., Tan, Y., & Mookerjee, V. (2011). Network effects: The influence of structural capital on open source project success. *MIS Quarterly*, *35*, 813–829.

Spradley, J. P. (2016a). *Participant observation*. Waveland Press.

Spradley, J. P. (2016b). *The ethnographic interview*. Waveland Press.

Stewart, K., & Ammeter, T. (2002). An exploratory study of factors influencing the level of vitality and popularity of open source projects. *ICIS 2002 Proceedings*, 88.

Stewart, K., & Gosain, S. (2001). An exploratory study of ideology and trust in open source development groups.

Subramaniam, C., Sen, R., & Nelson, M. L. (2009). Determinants of open source software project success: A longitudinal study. *Decision Support Systems*, *46*(2), 576–585.

Tajfel, H. (1974). Social identity and intergroup behaviour. *Information (International Social Science Council)*, *13*(2), 65–93.

Tajfel, H., & Turner, J. C. (1978). Intergroup behavior. In *Introducing social psychology* (pp. 401–466). Penguin Books.

Tashakkori, A., & Teddie, C. (1998). *Mixed methodology: Combining qualitative and quantitative approaches* (Vol. 46). Sage.

Teixeira, J., Mian, S., & Hytti, U. (2016). Cooperation among competitors in the open-source arena: The case of OpenStack. *ArXiv Preprint ArXiv:1612.09462*.

Terrell, J., Kofink, A., Middleton, J., Rainear, C., Murphy-Hill, E., Parnin, C., & Stallings, J. (2017). Gender differences and bias in open source: Pull request acceptance of women versus men. *PeerJ Computer Science*, *3*, e111.

Thomas, D., & Hunt, A. (2004). Open source ecosystems. *IEEE Software*, *21*(4), 89–91.

Thomas, R., Waldemar, H., Philipp, L., & Stefan, S. (2017). An empirical analysis of build failures in the continuous integration workflows of Java-based open-source software. In *proceedings of the 14th international conference on mining software repositories (MSR '17)* (pp. 345–355). IEEE Press.

Torkar, R., Minoves, P., & Garrigós, J. (2011). Adopting free/libre/open source software practices, techniques and methods for industrial use. *Journal of the Association for Information Systems*, *12*(1), 1–122.

Turner, J. C. (1975). Social comparison and social identity: Some prospects for intergroup behaviour. *European Journal of Social Psychology*, *5*(1), 1–34.

Turner, J. C., Brown, R. J., & Tajfel, H. (1979). Social comparison and group interest in ingroup favouritism. *European Journal of Social Psychology*, *9*, 187–204.

Van de Ven, A. H. (2007). *Engaged scholarship: A guide for organizational and social research*. Oxford University Press on Demand.

Van Maanen, J. (1988). *Tales of the field: On writing ethnography*. The University of Chicago Press.

von Krogh, G., Haefliger, S., Spaeth, S., & Wallin, M. W. (2012). Carrots and rainbows: Motivation and social practice in open source software development. *MIS Quarterly*, *36*, 649–676.

Vygotsky, L. S. (1964). Language and thought. *Annals of Dyslexia*, *14*(1), 97–98.

Walia, N., Rajagopalan, B., & Jain, H. (2006). Comparative investigation of vulnerabilities in open source and proprietary software: An exploratory study.

Walsh, G., Schaarschmidt, M., & Von Kortzfleisch, H. F. (2012). Harnessing free external resources: Evidence from the open source field.

Wang, X., & Lantzy, S. (2011). A systematic examination of member turnover and online community health. In *the proceedings of the 32nd international conference on information systems* (p. 25). ICIS.

Weick, K. E. (1989). Theory construction as disciplined imagination. *Academy of Management Review*, *14*(4), 516–531.

Wells, J. D., Valacich, J. S., & Hess, T. J. (2011). What signal are you sending? How website quality influences perceptions of product quality and purchase intentions. *MIS Quarterly*, *35*, 373–396.

Wynn, D., Boudreau, M.-C., & Watson, R. (2008). Resilience of professional open source ecosystems.

Xiao, X., Lindberg, A., Hansen, S., & Lyytinen, K. (2018). "Computing" requirements for open source software: A distributed cognitive approach. *Journal of the Association for Information Systems*, *19*(12), 2–1252.

Yamashita, K., McIntosh, S., Kamei, Y., Hassan, A. E., & Ubayashi, N. (2015). Revisiting the applicability of the pareto principle to core development teams in open source software projects. *Proceedings of the 14th International Workshop on Principles of Software Evolution*, 46–55.

Youssef, A., & Capiluppi, A. (2015). The impact of developer team sizes on the structural attributes of software. *Proceedings of the 14th International Workshop on Principles of Software Evolution*, 38–45.

Yu, L., Ramaswamy, S., & Bush, J. (2007). Software evolvability: An ecosystem point of view. *Third International IEEE Workshop on Software Evolvability*, *2007*, 75–80.

Yuan, Y. C., & Gay, G. (2006). Homophily of network ties and bonding and bridging social capital in computer-mediated distributed teams. *Journal of Computer-Mediated Communication*, *11*(4), 1062–1084.

Zargar, M. S. (2013). Reusing or reinventing the wheel: The search-transfer issue in open source communities.

Zgraggen, E., Zhao, Z., Zeleznik, R., & Kraska, T. (2018). Investigating the effect of the multiple comparisons problem in visual analysis. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12.

Zhang, Z., Yoo, Y., Wattal, S., Zhang, B., & Kulathinal, R. (2014). Generative diffusion of innovations and knowledge networks in open source projects.

AUTHOR BIOGRAPHIES

Kevin Lumbard is an Assistant Professor of Computer Science at Creighton University. Kevin's PhD is in Information Technology and Human-centered Computing. His work focuses on the sociotechnical aspects of design, which assumes both social *and* technical factors influence the design, functionality, and usage of technology. He is a former Web developer and a certified project manager. Kevin's research explores open source software development, corporate engagement with open source communities, and the health of open source communities. He uses engaged fieldwork and trace data analysis to research open source software communities and the

changing nature of technology work. Kevin's work has been published in CSCW, CAIS, HICSS, IDETC, IEEE Computer, and SoHeal. Kevin is a charter member, maintainer, and board member of the Linux Foundation's Community Health Analytics OSS Project (CHAOSS) https://chaoss.community/.

Matt Germonprez is the Mutual of Omaha Distinguished Chair of Information Science & Technology and Professor of Information Systems and Quantitative Analysis in the College of Information Science & Technology at the University of Nebraska Omaha. He uses qualitative field studies to research corporate engagement with open communities and the dynamics of work found in these engagements. His lines of research have been funded by numerous organizations including the Alfred P. Sloan Foundation, the Ford Foundation, the National Science Foundation, the Chan Zuckerberg Initiative, and Mozilla. Matt is the co-founder of the Linux Foundation Community Health Analytics OSS Project (CHAOSS). He has had work accepted at ISR, MISQ, JAIS, JIT, ISJ, I&O, CSCW, OpenSym, Group, HICSS, IEEE Computer, and ACM Interactions. Matt is an active open source community member, having presented design and development work at LinuxCon, the Open Source Summit NA, the Linux Foundation Open Compliance Summit, the Linux Foundation Collaboration Summit, and the Open Source Leadership Summit.

Sean Goggins is an Electrical Engineering and Computer Science Professor at the University of Missouri. His research foci are open-source software, serious game analytics, and human-centred data science. Sean is a founding member of the Linux Foundation's working group on community health analytics for open-source software (CHAOSS). His work is funded by the National Science Foundation, the Sloan Foundation, the Ford Foundation, Mozilla, the US Department of Education, Red Hat Software, and the US Navy's Office of Naval Research. Sean also created the Data Science and Analytics Master's programme at Missouri from 2013 to 2018. Sean's publications focus on understanding how social technologies influence organizational, small group, and community dynamics, typically including analysis of electronic trace data from systems combined with the perspectives of people whose behaviour is traced. He lives in Columbia, MO, with his wife Kate, two daughters (a third, the oldest, is off getting her Ph.D.), and a dog named Huckleberry.

How to cite this article: Lumbard, K., Germonprez, M., & Goggins, S. (2024). An empirical investigation of social comparison and open source community health. *Information Systems Journal*, *34*(2), 499–532. https://doi.org/10.1111/isj.12485

APPENDIX A: INTERVIEW PROTOCOLS

A.1 | Interview protocol 1

1. Why are you interested in community health metrics?
2. Could you talk about some specific instances where you have used community health metrics in the past?
3. Are you currently deploying any health metrics? If so, how are you applying health metrics?
4. Outside of your own deployment of metrics, are you aware of how health metrics are being used?
5. How could you see yourself contributing to the CHAOSS community?
6. There are several communities, projects, and organizations looking at community health. How familiar are you with some of these different communities, projects, and organizations?
   a. How similar are the discussions across these communities?
   b. What, if anything, is missing from these community discussions?

7. What do you believe the CHAOSS community adds to the discussion of community health metrics?
8. How do you believe insight about the design and use of metrics will be generated?
9. When you look at contributions to the metrics conversation coming from different sources, what factors determine the quality of those contributions?
10. A primary goal of CHAOSS is to develop technology and platform-agnostic metrics. As such, how do you think we can measure the success of metrics across communities?
11. Are there types of metrics that you believe are universal across OSS communities?
12. Do you have any concerns about the deployment of health metrics?

At this time, the CHAOSS community is exploring metrics related to project growth, maturity, and decline, project diversity and inclusion, project risk, and project dependency.

13. Of these metric areas, which one interests you the most?
    a. What are your thoughts on this metric area?
    b. What individual metrics could help in its representation?
        i. What could these metrics signal?
        ii. How could the metric be related to applied value (use case)?

A.2 | Interview protocol 2

1. When you contribute to open source projects, how often do you collaborate with other contributors on tasks versus performing tasks alone?
2. When you are working on an open source project, how aware are you of other contributors?
3. When you contribute to an open source project, how aware are you of overall project design goals?
4. What are the different kinds of relationships you have with your other collaborators on the project? Or, more aptly, do you have direct relationships with people or software-mediated relationships?
5. What important questions can you answer about your project? (What do you know about your project?)
6. What questions do you have about your project that you do not have answers to but would like?
7. How do you know your project is good? How do you define “good” for your project?
8. Is what you consider a “good” project the same as what others in your project's group consider a “good” project?
9. Are there other projects or multiple projects you seek for guidance with your project? Could you give me an example?
10. When comparing your projects, what factors help you determine if the comparison is relevant?
11. In what areas could another project influence your decisions? Do you use information about other projects to make decisions about project engagement?

Take a moment and think about potential use cases for community health metrics. These can be use cases that you have successfully used prior or even use cases that you believe would be helpful in the future.

12. Use Case Questions:
    a. What question are you trying to answer?
    b. What is the use case title?
    c. Why do you need this use case?
    d. What metrics may inform this use case?

APPENDIX B: CODING AND THEMATIC ANALYSIS

See Figures B1–B3.

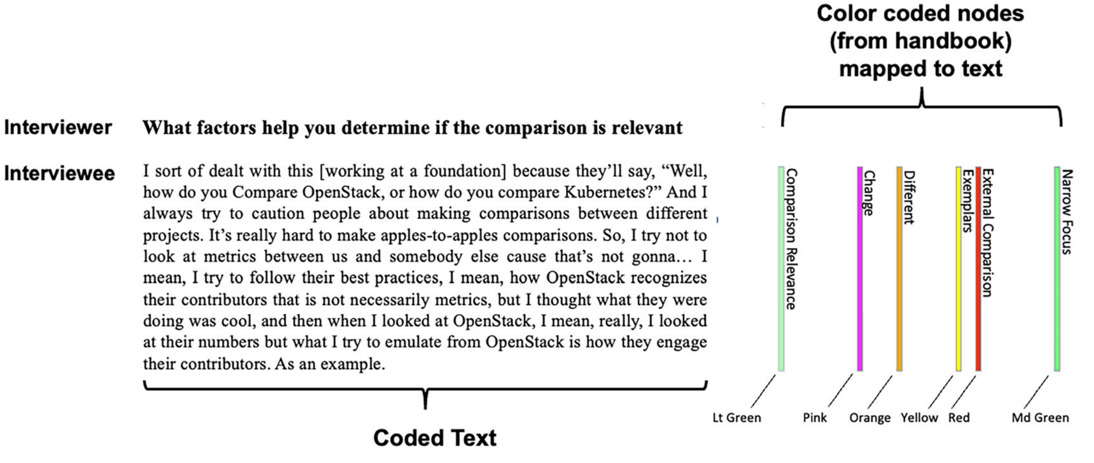

FIGURE B1 Example of NVivo code mapping.

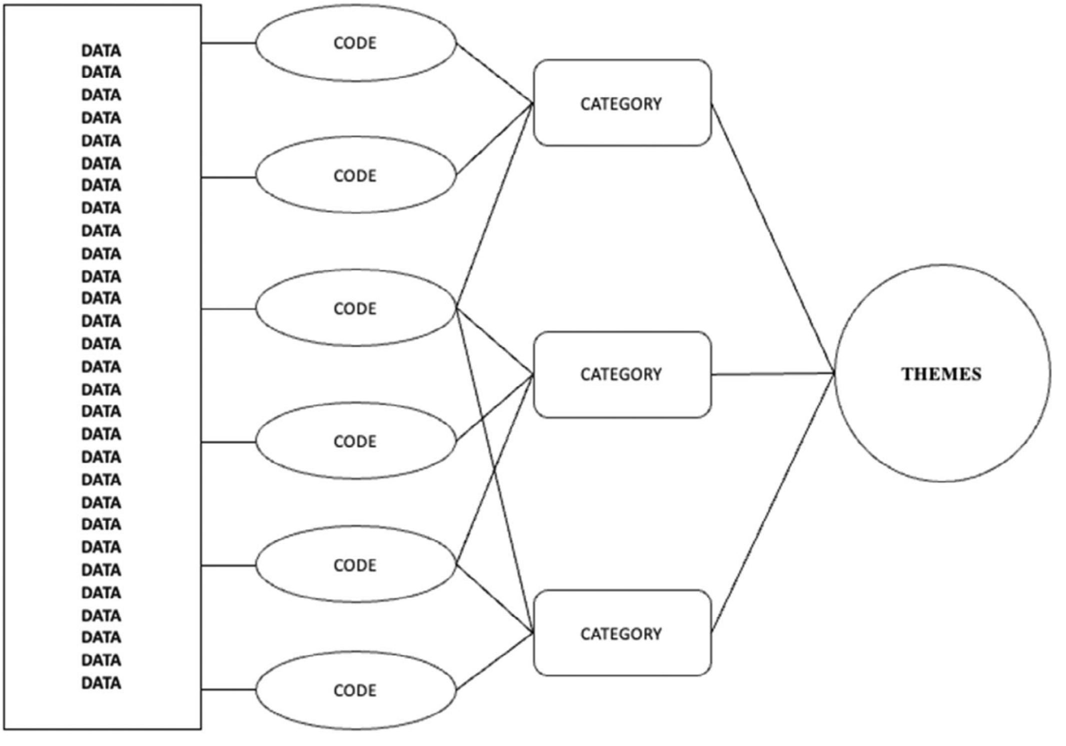

FIGURE B2 Thematic analysis procedure adapted from Miles et al. (2014).

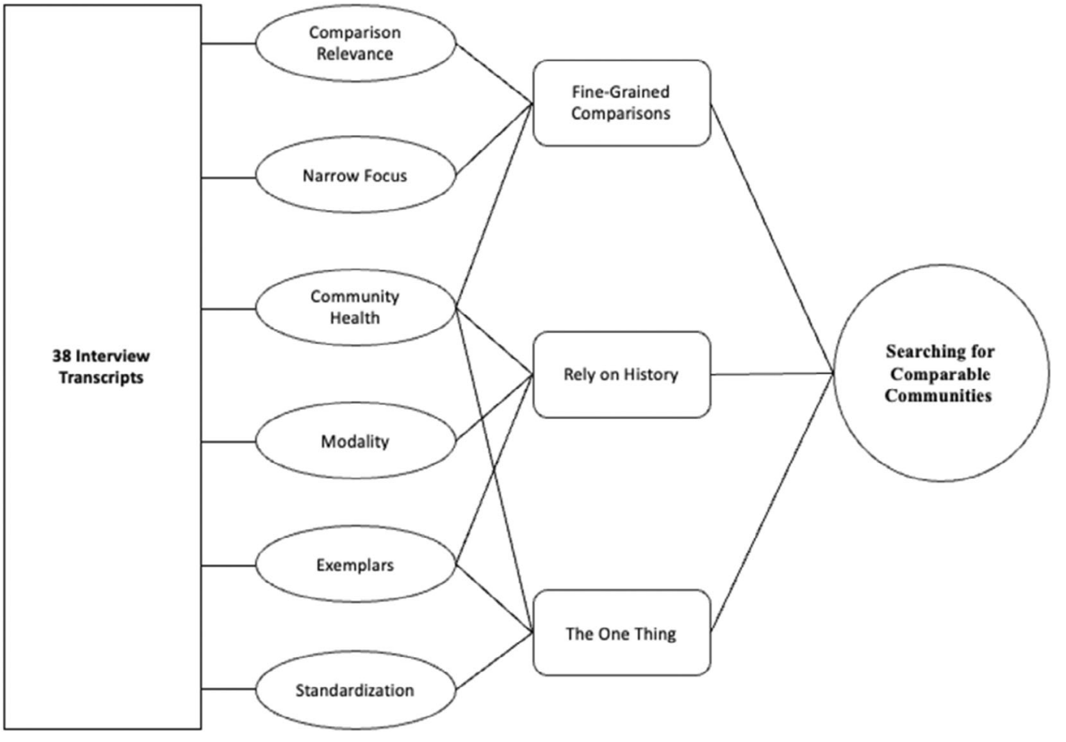

FIGURE B3 Construction of theme—Searching for comparable communities using Miles et al. (2014).