How to characterize the health of an Open Source Software project? A snowball literature review of an emerging practice

Johan Linåker
johan.linaker@ri.se
RISE Research Institutes of Sweden
Lund, Sweden

Efi Papatheocharous
efi.papatheocharous@ri.se
RISE Research Institutes of Sweden
Lund, Sweden

Thomas Olsson
thomas.olsson@ri.se
RISE Research Institutes of Sweden
Lund, Sweden

ABSTRACT

Motivation: Society's dependence on Open Source Software (OSS) and the communities that maintain the OSS is ever-growing. So are the potential risks of, e.g., vulnerabilities being introduced in projects not actively maintained. By assessing an OSS project's capability to stay viable and maintained over time without interruption or weakening, i.e., the OSS health, users can consider the risk implied by using the OSS as is, and if necessary, decide whether to help improve the health or choose another option. However, such assessment is complex as OSS health covers a wide range of subtopics, and existing support is limited. **Aim**: We aim to create an overview of characteristics that affect the health of an OSS project and enable the assessment thereof. **Method**: We conduct a snowball literature review based on a start set of 9 papers, and identify 146 relevant papers over two iterations of forward and backward snowballing. Health characteristics are elicited and coded using structured and axial coding into a framework structure. **Results**: The final framework consists of 107 health characteristics divided among 15 themes. Characteristics address the socio-technical spectrum of the community of actors maintaining the OSS project, the software and other deliverables being maintained, and the orchestration facilitating the maintenance. Characteristics are further divided based on the level of abstraction they address, i.e., the OSS project-level specifically, or the project's overarching ecosystem of related OSS projects. **Conclusion**: The framework provides an overview of the wide span of health characteristics that may need to be considered when evaluating OSS health and can serve as a foundation both for research and practice.

CCS CONCEPTS

• **Software and its engineering → Open source model**; • **Human-centered computing → Open source software**.

KEYWORDS

Open Source Software, Software Ecosystem, Health, Sustainability, Software Quality.

OpenSym 2022, September 7–9, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9845-9/22/09.
https://doi.org/10.1145/3555051.3555067

ACM Reference Format:
Johan Linåker, Efi Papatheocharous, and Thomas Olsson. 2022. How to characterize the health of an Open Source Software project? A snowball literature review of an emerging practice. In *The 18th International Symposium on Open Collaboration (OpenSym 2022), September 7–9, 2022, Madrid, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3555051.3555067

1 INTRODUCTION

Open Source Software (OSS) makes up a pivotal building block in today's digital infrastructure, both in industry and society at large. Due to the nature of OSS, organizations thereby to a large extent become reliant on the external maintenance of the different OSS projects carried out within their respective communities. If an OSS project is not actively maintained, the risk of vulnerabilities being introduced (either intentionally or not) may rise [55]. These can in turn spread with costly consequences within and between organizations, potentially causing harm to the whole business ecosystems and society in general (cf. Heartbleed and Log4Shell. Organizations, therefore, need to consider the health of the OSS projects that they use to manage the risk coupled with the usage, or dependence of thereof [12, 120].

With OSS health, we consider an *OSS project's capability to stay viable and maintained over time without interruption or weakening.* A topic that we find complex given the wide variety of subtopics included on the socio-technical spectrum (e.g., toxicity [41], sponsorships [95], marketing [88], diversity [47], badging [131], burnout [89], newcomer barriers [122]). Further complexity is introduced as OSS projects seldom can be considered in isolation due to complex dependency networks of up- and downstream projects [24, 132], implying that both the focal project and its overarching ecosystem of dependencies need to be considered.

Models and approaches have been developed by extant research to evaluate OSS projects from different perspectives, such as sourcing [75, 120], alliance partnerships [118], quality [141], and software ecosystem health [64]. These approaches often have a specific focus such as on source code quality [141], internal capabilities to consume the OSS [12], or the potential to build professional business relations through the community [118]. In contrast, there is still a lack of a holistic view of the wider area of research that OSS health implies (see e.g., [3, 50, 107, 122, 126]), as well as processes that can help organizations to evaluate OSS projects. Interest has, however, emerged around the topic, both in academia [1], industry [55], and community settings [2]. Of these, one notable exception exists in the case of the CHAOSS project [55], a community project collaboratively developing health metrics within themes such as value, risk,

[1] https://soheal.github.io/
[2] https://sustainoss.org/

or evolution. Within each theme, metrics are then provided under several focus areas, each with a specific goal that the metrics aim to provide answers to, e.g., to *"[l]earn about the types and frequency of activities involved in developing code"* under the focus area Code Development Activity within the Evolution theme[3]. No instructions are provided on which metrics to choose. This is instead up to each user to decide.

Extant research does, however, not provide an overview of this wide area of research that OSS health implies (see e.g., [3, 50, 107, 122, 126]), even though community and industry-oriented initiatives are starting to emerge [55]. In this study, our goal is to address this gap by initializing the development of an assessment framework that can provide a comprehensive overview of the health of an OSS project. Specifically, we seek to answer the question *what characteristics affect the health of an OSS project?*

To address the question, we conducted a snowball literature review [140] with a start set of 9 highly cited papers [3, 26, 50, 52, 64, 107, 122, 126, 132] after which 93 papers were identified in the first iteration, and 53 additional in a second. We structurally coded the characteristics, related metrics, and underpinning purpose for a paper's analysis based on an a-priori framework derived from the ecosystem health literature [64, 82]. Through axial coding, we then designed a framework consisting of 107 health characteristics among 15 themes.

2 RESEARCH DESIGN

We conducted a literature review to answer the research question on OSS health characterization using a snowballing search strategy [140]. Using a start set of papers, snowballing activity is conducted backward and forwards in iterations. In the backward snowball search, references from a previously included paper in the start set are each reviewed by their title, and place and context in the paper where the reference is used. In a forward snowball search, Google Scholar is used to identify papers citing a previously included paper. Papers are reviewed by their title, followed by their abstracts, and a full read, incrementally until a decision can be made on whether the paper should be included or not.

Typically, the pattern continues until no new papers are identified [140]. In this study, however, the goal is not to conduct a systematic literature review. Rather, we aim to collect a comprehensive knowledge base that can provide an understanding of what characteristics affect the health of an OSS project (RQ), and serve as a foundation for an initial design of an artifact that, iteratively, can be validated and improved in future work through empirical research. Hence, we favor the saturation of characteristics rather than new papers. Below, we describe our research design and process in further detail.

2.1 Inclusion and Exclusion Criteria

In line with our definition of OSS health, along with guidance from related work [64, 83], we defined the following inclusion and exclusion criteria (denoted IC and EC respectively):

- IC Papers on the growth, attraction, or retention of contributors to OSS projects.
- IC Papers on the maintainers' ability to maintain their OSS projects.
- IC Papers on the evaluation of quality or risks of an OSS project related to its health.
- IC Papers explicitly mentioning the keywords [64, 83] "health", "sustain*", "propensity", "longevity", or "survival" in the context of OSS in the title, abstract, introduction or conclusion section.
- EC Not explicitly focused on OSS or focused on (potential) contributors' motivations to engage in an OSS project.
- EC Published before 2012 and not available in full text.
- EC Grey and white literature, including book chapters, reports, and student theses.
- EC Idea and opinion papers, extended abstracts, duplicate studies, secondary studies.
- EC For extension works most recent published work is included, and others excluded.
- EC Non-English papers.

[3]https://github.com/chaoss/wg-evolution

Startset	First forward	Total (excl. dupl.)	Tentatively included	Included	Second forward	Total (excl. dupl.)	Tentatively included	Included
8	601	1206	150	93	1901	2386	139	53
	First backward				Second backward			
	708				658			

Figure 1: Overview of the number of papers identified and finally included in the first and second iteration of forward and backward snowball searches. Numbers presented in the Total columns are excluding duplicates.

2.2 Start set

As a first step, a start set of papers was to be identified as a baseline for the snowballing iterations. Based on keyword searches using Google Scholar it was found that "health", "sustainability", "sustain" and "survival" were used interchangeably in alignment with our previously defined definition of OSS health. A search string was constructed accordingly and contextualized to reflect both general and "ecosystem"-focused literature:

("open source" OR "open-source") AND ("project" OR "ecosystem") AND ("health" OR "sustainability" OR "sustain" OR "survival")

This search string was applied through Google Scholar to gain non-biased recommendations to any specific publishing venue [140]. From the search, three studies were identified as highly cited and seminal papers in the area [52, 64, 132] among the first 20 results presented, all passing defined criteria.

Due to the rather wide definition of OSS health, which is further emphasized by Manikas and Hansen [83], we decided to specifically look for systematic reviews that may reflect different aspects or areas of the OSS health literature. We, therefore, again using Google Scholar, applied the adapted search string:

("open source" OR "open-source") AND ("project" OR "ecosystem") AND ("health" OR "sustainability" OR "sustain" OR "survival") AND "systematic" AND ("review" OR "examination")

Among the top 20 presented results, five reviews were identified as relevant secondary studies [3, 26, 50, 107, 122]. As they are secondary studies, they are not included in our analysis and only make up a starting point to find primary studies relevant to our study.

2.3 First Iteration

In the first iteration, the first author conducted a forward and backward snowball search based on the previously defined start set. The data collection was originally performed in January 2022 but repeated in correlation with the second iteration between the 22nd to 25th of May 2022 to identify recent publications. The (overall) search resulted in an initial set of 150 tentatively included papers from a total of 1206 papers (see Fig. 1).

In the following step, all three authors independently reviewed a sample of 10 percent (i.e., 15 papers) of the tentatively included papers to decide if papers would be included or not. In the same process, the authors also extracted data where applicable. After the independent review process, all three authors discussed and compared findings to arrive at a common understanding of the interpretation and final application of the inclusion and exclusion criteria, as well as the performance of the data extraction process. We continued by separately reviewing and coding papers based on the *a-priori* codes: *Purpose (P)* for analyzing or discussing the health of an OSS project; *Characteristics (C)* that affects or reflects the health of an OSS project; *Metrics (M)* that can quantify, or qualitatively describe, a characteristic.

During the continued review and coding process, two additional negotiations were performed in terms of agreement of inclusion and coding, each time picking a 10 percent random sample with five papers coded from each of the three authors. This rendered in an overall sample of 30 percent (i.e., 45 papers) being co-analyzed by the three authors. Overall agreement was found on all three occasions with minor adjustments coming out as an effect of the negotiations, e.g., whether a certain extracted element should be considered a purpose, characteristic or metric, or whether a paper should be included or excluded based on the criteria. In the end, 93 papers were included and coded in the first iteration.

Following, we performed a structured coding of the characteristics based on an a-priori-defined health framework inspired by the work of Manikas and Hansen [83] and Jansen [64]. The framework consists of two dimensions; the level of abstraction, and the socio-technical dimension.

- Level of abstraction
  - *Network-level* concerns characteristics related to the overarching software ecosystem or network that the OSS project is part of, e.g., a language-specific package ecosystem as NPM which in turn consists of multiple OSS components, or the OpenStack ecosystem which in turn consists of numerous of integrating sub-module OSS projects.
  - *Project-level* concerns characteristics focused explicitly on the OSS project.
- Socio-technical dimension
  - *Actors* concerns the community of developers and users that is part of the OSS project or its overarching software ecosystem.
  - *Software* concerns the OSS and related artifacts (e.g., documentation) that is being developed by the community of actors, or individuals, that are either part of the OSS project or its overarching ecosystem.
  - *Orchestration* concerns the governance exercised in terms of development, collaboration, and usage of the software by its community of actors, either within an OSS project or its the overarching ecosystem.

After the structural coding, an axial coding process was performed within each of the six categories of codes that follows by the two dimensions.

2.4 Second iteration

In the second iteration, the first author performed the forward snowball, while the third author performed the backward snowball, each resulting in 71 and 68 tentatively included papers out of 1901 and 658 papers respectively. The data collection was performed from the 22nd to the 25th of May 2022. After a second review, 53 papers were included making a total of 146 papers, also considering the first iteration, distributed between 2012-2022. The 53 papers were coded using the structured and axial coding process as the first iteration.

In the axial coding, the code book with themes and characteristics generated from the first iteration, structured as per the a-priori framework, was used as a foundation. The coding process rendered in slight modifications, most prominently with the emergence of a new theme focused on security aspects. On the general level, however, we experienced a saturation in the type of characteristics of health that appeared why we decided to stop.

The final framework was verified through peer debriefing and discussions including all three authors and is further presented in the following section.

3 RESULTS

The assessment framework consists of **107** health characteristics divided among 15 themes each providing different perspectives on OSS health (see tables 1, 2 and 3). Below we provide a summary per theme along with an overarching question contextualizing how the underpinning characteristics affect the health of an OSS project. Codes are provided in parenthesis per characteristic (e.g., a-com-1, meaning [actors-category] - [communication-theme] - and [first characteristic] in the alphabetical order) connects to what papers are related to the characteristic as presented in tables 1, 2 and 3. We refer readers to the supplementary material to explore metrics related to the characteristics as identified in the literature, and to investigate the framework in-depth [78].

3.1 Actors-oriented characteristics

Communication: With communication, we consider the social interactions internally between the actors within an OSS community, and externally by the community in their outward-facing communication. They help to answer the question on *how productive an OSS project is in planning and discussing the evolution and development of its technical and non-technical deliverables.*

Literature highlights several ways in which the communication takes place, such as mailing lists, issue trackers, and pull-requests. The responsiveness shown by the community through these channels was a commonly referred to characteristic, both considering

Table 1: Overview of actors-oriented health characteristics per theme. Characteristics listed under unique identifiers linking to the respective codes in the online supplementary material [78], P = project level, and N = network-level focus.

Actors / Communication		
a-com-1	Response-quality	P: [138], [121] [58] [75]
a-com-2	Response-time	P: [10], [55], [121], [23], [118], [58], [75], [67], [123], [22]
a-com-3	Social activity	P: [86], [138], [24], [58], [75], [12], [67] N: [92]
a-com-4	Visibility	P: [48], [49], [41], N: [92]
Actors / Culture		
a-cult-1	Conflicts	P: [20], [58], [45]
a-cult-2	Contributor satisfaction	P: [58], [59]
a-cult-3	Contributor recognition	P: [136]
a-cult-4	Language heterogeneity	P: [58]
a-cult-5	Openness	P: [121], [23], [118], [58] [103], [75]. [22]
a-cult-6	Sentiment	P: [129], [53], [121], [106], [35], [58], [88], [43], [123]
Actors / Finance		
a-fin-1	Financial stability	P: [90], [138], [48], [71], [49]
a-fin-2	Financial support	P: [138], [95], [71], [147], [135], [110]
Actors / Diversity		
a-div-1	Application diversity	P: [136] N: [92]
a-div-2	Demographic diversity	P: [104], [48], [49], [18], [72], N: [47]
a-div-3	Organizational diversity	P: [105], [136], [36]. [37], [132], [64], [12], N: [25], [100], [64]
a-div-4	Target users	P: [138], [113]
a-div-5	Technical knowledge	P: [104]
Actors / Popularity		
a-pop-1	Competing projects	P: [20], [132]
a-pop-2	End-user popularity	P: [87], [136], [103], [145]
a-pop-3	External community interest	P: [131], [148], [94], [136], [75], [109], [11], [145], [85], [12], [65], [67], [51]
a-pop-4	Project popularity	N: [136], [92]
a-pop-5	SDG	P: [59]
a-pop-6	Size	P: [75]
a-pop-7	Technical adoption	P: [145]
Actors / Stability		
a-stab-1	Age	P: [94]
a-stab-2	Attrition	P: [108], [79]
a-stab-3	Ecosystem growth	N: [100]
a-stab-4	Forks	P: [112], N: [20], [23]
a-stab-5	Growth	P: [133], [108], [48], [49], [79]
a-stab-6	Knowledge concentration	P: [6], [111], [48], [13], [23], [49], [5], [44], [42], [99]
a-stab-7	Life-cycle stage	P: [113], [20]
a-stab-8	Predicted evolution	P: [76], [31], [142], [74], [98], [81], [12]
a-stab-9	Project growth	N: [133], [100], [92]
a-stab-10	Retention	P: [77], [108], [115], [124], [55], [36], [35], [89], [39], [9], [38], [73], [144]
a-stab-11	Size	P: [48], [49], [36], [135], [135], [12], N: [91], [92]
a-stab-12	Turnover	P: [46], [104], [146]
Actors / Technical activity		
a-tech-1	Contributors' development activity	P: [21], [58], [75]
a-tech-2	Efficiency	P: [131], [113], [121], [19], [27], [54]
a-tech-3	Maintainers' development activity	P: [23], [139], [75], [132]
a-tech-4	Non-code contributions	P: [15], [130]
a-tech-5	Overall development activity	P: [87], [133], [90], [97], [113], [24], [94], [136], [101], [21], [48], [71], [49], [118], [66], [36], [103], [75], [137], [145], [4], [68], [134], [12], [132], [67], N: [7], [93], [25], [100], [47], [64], [92]

response time (a-com-2), and *response quality* (a-com-2), e.g., in terms of the level of detail, complexity, and correctness. Another characteristic adding to the response-time is the general *social activity* (a-com-3), or frequency, in the communication of the OSS project, e.g., the number of issues opened or comments posed in a certain time interval. Outward-facing communication and *visibility* (a-com-4) was another aspect considered, e.g., how active the community is in terms of social media presence.

Culture: Cultural characteristics help to answer the question *how able a community is to facilitate a positive and inclusive collaboration and dialogue among existing and potential actors.* They further help to contextualize the social behavior and norms expressed and experienced by the individuals present in a community. The literature emphasizes the experience of contributors, especially in terms of experiencing a personal *satisfaction* (a-cult-2), and being *recognized* for their contributions (a-cult-3), independent of the contribution type and complexity. The presence of *conflicts* (a-cult-1) and how these are managed by a community, as well as the general *openness* (a-cult-5) in terms of the community's mindset in welcoming and encouraging contributions, inputs, and questions are also highlighted by several studies. The general *sentiment* and tone (a-cult-6) in the communication is another commonly referred to characteristic in literature, where the presence of negative (e.g., insulting, entitled, arrogant, trolling, or unprofessional) and positive qualities (friendliness, welcoming, inclusion) was investigated. *Language heterogeneity* (a-cult-4), or rather the lack of a common language in a community, was a specific concern, investigated by one study, that may cause cross-communication hurdles.

Diversity: Diversity-related characteristics describe the OSS project's or its overarching ecosystem's ability to be receptive to diversity and self-renew itself, thereby helping to answer the question *how able a community is to accommodate and attract a diverse community of actors, while enabling existing and new use cases of the OSS project.* Among the literature surveyed, multiple angles were covered. One study highlighted the aspect of diversity in the use cases and applications based on the OSS project among its users (a-div-1), aligning with the more general organizational diversity (a-div-3) aspect considering e.g., the size, location, financial stability, business model and influence of the organizations engaged in the community. Demographic diversity (a-div-2), another multifaceted aspect was also investigated, e.g., in terms of gender, culture, and geographical situation. The level and diversity regarding the technical knowledge (a-div-5), e.g., considering programming languages, among individuals was also raised as the diversity among the target users (a-div-4) of the OSS project, e.g., tech savvy or general users.

Finance: Finance-related characteristics describe the financial support (a-fin-2) in terms of funding and sponsorship provided to or accepted by the OSS community, and the general financial stability (a-fin-1) of the actors in the community that are maintaining or contributing to the OSS project. These characteristics thereby help to answer the question *how financially viable actors are in an OSS community in terms of being able to dedicate their time and resources to the long-term maintenance of the OSS project.*

Popularity: Characteristics related to popularity describe the general external interest in the OSS project or its overarching ecosystem, helping to answer the question *how popular and well-adopted an OSS project is among existing and potential end-users and contributors.* End-user popularity (a-pop-2), i.e., the level of interest displayed in the project by its consumers, and the external community interest (a-pop-3) shown towards the OSS project were the two most highlighted characteristics in this theme. On an ecosystem-level, one study highlighted the general popularity of the projects hosted within the ecosystem (a-pop-4). The current size of the OSS community in terms of users and developers (a-pop-6), any connection between the OSS project's use case(s) and the Sustainable Development Goals (a-pop-5), along with the technical inclusion and adoption of an OSS project in downstream software implementations were other characteristics highlighted (a-pop-7). The presence of competing projects was further emphasized as a characteristic that may affect the popularity (a-pop-1).

Stability: Characteristics related to stability describe the resilience and robustness of the OSS community or its overarching ecosystem in terms of their population, helping to answer the question *how capable the OSS project is in terms of preserving a critical population of actors with the capability to maintain the OSS project long-term.* The growth (a-stab-5), retention (a-stab-10), attrition (a-stab-2), and overall turnover (a-stab-12) and size (a-stab-11) of users and developers of an OSS project are characteristics thoroughly investigated by literature. A related characteristic also thoroughly studied is the concentration or distribution of contributions and knowledge to certain individuals or groupings within an OSS project, commonly quantified and described through the bus- or truck factor of a community (a-stab-6). Some studies focused on characterizing OSS projects in terms of their state from a life-cycle perspective (a-stab-7), while others were more forward-looking and focused on predicting future development activity in the OSS project, e.g., in terms of growth or dormancy (a-stab-8).

Technical activity: The technical activity covers characteristics describing the overall technical activity, helping to answer the question *how productive an OSS project is in evolving and developing its technical and non-technical deliverables.* The theme may, in contrast to the communication theme be considered as the technical pulse of a community and a sign of its productivity, both in terms of technical and non-technical contributions towards the evolution of the concerned OSS project. As per the literature, the technical activity can be considered and evaluated both from the maintainers' (a-tech-3), contributors' (a-tech-1), and overall community perspective (a-tech-5). Effectiveness and ease of an OSS project in managing and moving the development forward, e.g., in accepting and reviewing issues and pull-requests, is also highlighted as an important aspect (a-tech-2). Studies also highlight the importance of evaluating the activity in terms of non-code contributions specifically (a-tech-4).

3.2 Software-oriented characteristics

Development process: Characteristics relating to the development process describe the quality and formality of the processes and practices for how the development is performed, addressing the question *how capable a community is in terms of its development process to maintain the OSS project to a high quality long-term.* The most highlighted characteristic concerns how the onboarding of newcomers to the project is performed, e.g., in terms of mentorship, the introduction of newcomers, and listing of good issues to start

Table 2: Overview of software-oriented health characteristics per theme. Characteristics listed under unique identifiers linking to the respective codes in the online supplementary material [78], P = project level, and N = network-level focus.

Software / Documentation		
s-doc-1	Completeness	[71], [121], [23], [118], [12], [22]
s-doc-2	Complexity	[121], [73]
s-doc-3	Currentness	P: [131], [121], [58], [123]
s-doc-4	Development docs	P: [20], [121], [58], [103], [75], [22], [123]
s-doc-5	General docs	P: [20], [103], [75]
s-doc-6	Language availability	P: [58]
Software / Development process		
s-dev-1	Coding conventions	P: [121], [75], [137]
s-dev-2	Contribution process	P: [121], [58], [73]
s-dev-3	Coordination	P: [150]
s-dev-4	On-boarding	P: [125], [14], [80], [10], [121], [23], [103], [59], [8], [123], [73], [40]
s-dev-5	Process maturity	P: [48], [13], [49], [137], [123]
s-dev-6	Quality assurance	P: [131], [20], [71], [121], [103], [75], [22]
s-dev-7	Support	P: [12], [65], [62]
Software / License		
s-lic-1	Flexibility	P: [71]
s-lic-2	Implications	P: [117], [138], [113], [20], [118], [135], [12], [132], [114]
s-lic-3	Legal jargon	P: [58]
s-lic-4	Management	P: [58]
Software / General factors		
s-gen-1	Age	P: [117]
s-gen-2	Application domain	P: [113]
s-gen-3	Platform support	P: [117], [75]
s-gen-4	Project complexity	P: [121], [137], [22]
s-gen-5	Project independence	P: [75]
s-gen-6	Standards compliance	P: [75]
s-gen-7	Type of technologies	P: [117], [20], [10], [58], [75], [22]
Software / Scaffolding		
s-scaff-1	Build environment	P: [121], [60], [123]
s-scaff-2	Continuous integration	P: [20], [22]
s-scaff-3	Conversation history	P: [58]
s-scaff-4	Infrastructure accessibility	P: [121], [118], [58], [103], [132]
s-scaff-5	Infrastructure availability	P: [103]
Software / Security		
s-sec-1	Dependencies	P: [71], [121]
s-sec-2	Dependency management	P: [131], [34], [75], [32], [63], [56]
s-sec-3	Security practices	P: [119], [75]
s-sec-4	Trustworthiness	P: [75], [12]
s-sec-5	Vulnerability persistence	P: [33], N: [102], [1]
s-sec-6	Vulnerability presence	P: [28], [137], [102], [1]
Software / Technical quality		
s-tech-1	Architecture quality	P: [121], [75], [22], N: [25]
s-tech-2	Component quality	P: [90], [84], N: [90]
s-tech-3	Contribution quality	N: [47]
s-tech-4	Ease of integration	P: [71]
s-tech-5	Maintainability	P: [70], [20], [23], [132], [69], [57]
s-tech-6	Modularity	P: [138], [150], [75], [135]
s-tech-7	Other non-functional requirements	P: [67], [75]
s-tech-8	Product quality	P: [90], [138], [49], [118], [65]
s-tech-9	Source-code complexity	P: [94], [71], [121], [58], [75], [137], [135]
s-tech-10	Source-code quality	P: [116], [121], [118], [58], [103], [75], [146], [22], [134]

with (s-dev-4). A relating characteristic concerns the presence and quality of a contribution process, i.e., how contributions should be made, reviewed, managed, and merged in the OSS project (s-dev-3). Quality and maturity of processes and practices related to quality assurance (s-dev-6), coordination (s-dev-3), coding conventions (s-dev-1), and the development overall (s-dev-5) were also highlighted by different studies. From (primarily) a commercial user perspective, the extent, and quality of any support services provided by the OSS project or the actors engaged in or hosting the project, were also lifted (s-dev-7).

Documentation: Documentation-related characteristics describe the quality of general and technical documentation, addressing the question *how capable a community is to develop, persist, and disseminate knowledge among current and future actors engaged in the project.* General documentation encompasses documentation of general nature aimed at both the community, users, and others interested in the OSS project, e.g., readme, homepage, and user manuals (s-doc-5). Technical documentation refers to documentation covering different aspects of the development process, e.g., in terms of onboarding, planning, contributions, code comments, and quality assurance (s-doc-4). Certain characteristics focus explicitly on quality aspects of the documentation in general, including the completeness (s-doc-1), currentness (s-doc-3), as well as its level of complexity, and ease of understanding (s-doc-2). One study highlighted the availability of multi-lingual documentation (s-doc-6).

General characteristics: A special group of characteristics looks at *how attractive an OSS project is based on its general technical features.* These characteristics include general user aspects such the application domain, or product category, of the OSS (s-gen-2), the type(s) of platforms and operating system(s) that the OSS is intended for (s-gen-3), its age (s-gen-1), compliance with externally defined standards (s-gen-6), and independence of external software components (s-gen-5). Other characteristics are more technical, such as the OSS project's size and complexity (s-gen-4), and choice of programming languages, libraries, frameworks, and protocols (s-gen-7).

License: License-related characteristics were emphasized by several studies, highlighting *how license choices and related practices may affect the popularity and attractiveness of an OSS project, both for commercial actors and individuals.* One study highlights whether there is flexibility in terms of choosing between licenses for the OSS project (s-lic-1). Most studies, however, emphasize the importance of the implications of the license on e.g., redistribution, usage, and packaging (s-lic-2). On a more general level, the quality and presence of practices and processes for license management in the OSS project were emphasized as important for commercial actors (s-lic-4), while the presence of legal jargon was highlighted as a barrier to entry, especially for newcomers (s-lic-3).

Scaffolding: The scaffolding theme concerns *how robust and accessible the development and communication infrastructure used in the OSS project is in terms of enabling a collaborative and high quality maintenance of the project.* This includes both the availability (s-scaff-5), and accessibility and user-friendliness of tools used for communication and development in the OSS project (s-scaff-4). The presence and quality of continuous integration infrastructure, automation, and practices in the OSS project were also highlighted as important characteristics in terms of software quality and the general attractiveness of a project (s-scaff-2). The ease of setting up the build environment and compiling the OSS project is considered an important aspect to enable newcomers and lower the barrier to adoption of the OSS project (s-scaff-1).

Security: Characteristics in the security theme help answer the question *how robust an OSS project is in terms of mitigating and managing vulnerabilities and security-related aspects in the current and future maintenance of the project.* More specifically, studies have highlighted past, current, and future (predicted) presence of vulnerabilities and characteristics thereof in dependencies of an OSS project as an important characteristic (s-sec-6). So also the address and persistence of past and current vulnerabilities in an OSS project (s-sec-5). Practices relating to security (s-sec-3), and specifically in terms of dependency management (s-sec-2), e.g., in regards to managing "conflicting versions of nested dependencies" as well as updates and security patches, were also raised.

Technical quality: Technical quality is a rather wide theme considering both the OSS project in general and its code base specifically, helping to answer the question *how robust an OSS project is in terms of its technical quality, considering both a user and developer perspective.* Quality was highlighted both in terms of the product (s-tech-8), component (s-tech-2), architecture (s-tech-1), and source code level (s-tech 9). The complexity of the source code was specifically highlighted in several studies, both in terms of attracting and enabling developers to understand and contribute to the code base, but also in terms of potential correlations to the presence of bugs, vulnerabilities, and negative impact on quality requirements in general. Although several quality requirements were highlighted individually, modularity (s-tech-6), and maintainability (s-tech-5), i.e., the ease of maintaining the source code of the OSS project), were the two that received extra attention.

3.3 Orchestration-oriented characteristics

Orchestration: The orchestration-theme covers characteristics describing the governance structure and quality of the leadership, helping to answer the question of *how mature and open the orchestration is in the OSS project or its overarching ecosystem in terms of enabling an open and inclusive collaboration and long-term maintenance of the OSS project.* Explicitness, formality, and general recognition of the governance structure and leadership were especially highlighted (o-orch-4). As was the way in which the individuals in an OSS community are connected, collaborate, and grouped, explored primarily through the concepts of community patterns and community smells (o-orch-1). The same dimension concerns the overarching ecosystem in how communities collaborate to create resilience and synergies between each other (o-orch-2). Other characteristics regard the leadership's openness to input and transparency of discussions with actors engaged or with an interest in the OSS project (o-orch-6).

4 DISCUSSION AND CONCLUSIONS

Evaluating the health of an OSS project is a complex exercise. Knowing what to look for, and how to measure it may get out of hand due to a wide focus, or risk becoming too narrow-minded so that important aspects are missed. In this study, we set out to create an overview of the wide range of sub-topics related to OSS health.

Table 3: Overview of orchestration-oriented health characteristics per theme. Characteristics listed under unique identifiers linking to the respective codes in the online supplementary material [78], P = project level, and N = network-level focus.

Orchestration / Orchestration		
o-orch-1	Community structure	P: [16], [128], [29], [30], [58], [2], [96], [17], [61], [127]
o-orch-2	Ecosystem structure	N: [138], [48], [49], [132], [92]
o-orch-3	Explicitness of ecosystem	N: [7]
o-orch-4	Governance	P: [149], [138], [71], [118], [75], [135], [12], [12], N: [7]
o-orch-5	Information consistency	N: [48], [49]
o-orch-6	KPI-programme	N: [7]
o-orch-7	Openness	P: [118], [58]
o-orch-8	Processes	P: [143], [20], [23], [75]
o-orch-9	Trustworthiness	N: [48], [49]

Based on a snowball study over two iterations, including 146 primary studies, we derive a framework that consists of 107 health characteristics divided among 15 themes. The themes are dispersed over the socio-technical spectrum with the least coverage in terms of orchestration-related characteristics. It may further be noted that a limited portion of the characteristics is observed on the network level. This relates to the context of the studies included, whether they have focused on an ecosystem (i.e., network) perspective, or the OSS project more specifically. The identified studies confirm, however, the importance of not analyzing an OSS project in isolation. Its dependencies and ties to other projects play an important part, e.g., in terms of resilience and security.

Giving a detailed presentation of the whole framework, including all its characteristics and metrics is beyond the scope and format of this paper, which is why we refer readers to the supplementary material to investigate and explore the assessment framework in-depth [78].

Similar to the CHAOSS project, our framework provides limited guidance in terms of which characteristics to consider, and how. Specifically, we provide limited support in regards to what metrics to operationalize for each characteristic. Readers have to consider metrics as presented through the audit trail and code structure provided in the supplementary material [78]. In future research, we aim to address this gap through further iterations to design a more mature framework with related processes that can be tailored based on organizational context and requirements. We aim to leverage case studies, interview surveys and observations of health assessments.

Regarding the limitations in general, it should be noted that we do not claim to have systematically surveyed the literature. Rather, we have made design choices that have limited the search scope and potentially excluded papers (and characteristics) that might be of relevance. We do believe, however, that the snowballing approach has provided a broad sample of the literature, where we could observe a saturation in elicited characteristics.

REFERENCES

[1] Mahmoud Alfadel, Diego Elias Costa, and Emad Shihab. 2021. Empirical analysis of security vulnerabilities in python packages. In *2021 IEEE Int. conf. on Software Analysis, Evolution and Reengineering* (Honolulu, HI, US). IEEE, 446–457.

[2] Nuri Almarimi, Ali Ouni, Moataz Chouchen, and Mohamed Wiem Mkaouer. 2021. csDetector: an open source tool for community smells detection. In *Proc. of the 29th ACM Joint Meeting on European Software Engineering conf. and Symposium on the Foundations of Software Engineering* (Athens Greece). ACM, New York, NY, USA, 1560–1564.

[3] Carina Alves, Joyce Oliveira, and Slinger Jansen. 2018. Understanding Governance Mechanisms and Health in Software Ecosystems: A Systematic Literature Review. In *Enterprise Information Systems*, Slimane Hammoudi, Michał Śmiałek, Olivier Camp, and Joaquim Filipe (Eds.). Springer Int. Publishing, Cham, 517–542.

[4] Hirohisa Aman, Aji Ery Burhandenny, Sousuke Amasaki, Tomoyuki Yokogawa, and Minoru Kawahara. 2017. A Health Index of Open Source Projects Focusing on Pareto Distribution of Developer's Contribution. In *2017 8th Int. Workshop on Empirical Software Engineering in Practice* (Tokyo, Japan). IEEE, 29–34.

[5] Guilherme Avelino, Eleni Constantinou, Marco Tulio Valente, and Alexander Serebrenik. 2019. On the abandonment and survival of open source projects: An empirical investigation. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (Porto de Galinhas, Brazil). IEEE, 1–12.

[6] Guilherme Avelino, Leonardo Passos, Andre Hora, and Marco Tulio Valente. 2016. A novel approach for estimating truck factors. In *2016 IEEE 24th Int. conf. on Program Comprehension* (Austin, TX, USA). IEEE, 1–10.

[7] Alfred Baars and Slinger Jansen. 2012. A Framework for Software Ecosystem Governance. In *Software Business*, Michael A. Cusumano, Bala Iyer, and N. Venkatraman (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 168–180.

[8] Sogol Balali, Umayal Annamalai, Hema Susmita Padala, Bianca Trinkenreich, Marco A Gerosa, Igor Steinmacher, and Anita Sarma. 2020. Recommending tasks to newcomers in oss projects: How do mentors handle it?. In *Proc. of the 16th Int. Symposium on Open Collaboration* (Spain). ACM, New York, NY, USA, 1–14.

[9] Lingfeng Bao, Xin Xia, David Lo, and Gail C Murphy. 2019. A large scale study of long-time contributor prediction for github projects. *IEEE Transactions on Software Engineering* 47, 6 (2019), 1277–1298.

[10] Shahab Bayati. 2019. Effect of Newcomers' Supportive Strategies on Open Source Projects Socio-Technical Activities. In *2019 IEEE/ACM 12th Int. Workshop on Cooperative and Human Aspects of Software Engineering* (Montréal, QC, Canada). IEEE, 49–50.

[11] Hudson Borges and Marco Tulio Valente. 2018. What's in a github star? understanding repository starring practices in a social coding platform. *Journal of Systems and Software* 146 (2018), 112–129.

[12] Simon Butler, Jonas Gamalielsson, Björn Lundell, Christoffer Brax, Anders Mattsson, Tomas Gustavsson, Jonas Feist, Bengt Kvarnström, and Erik Lönroth. 2022. Considerations and challenges for the adoption of open source components in software-intensive businesses. *Journal of Systems and Software* 186 (2022), 111152.

[13] Simon Butler, Jonas Gamalielsson, Björn Lundell, Christoffer Brax, Johan Sjöberg, Anders Mattsson, Tomas Gustavsson, Jonas Feist, and Erik Lönroth. 2019. On company contributions to community open source software projects. *IEEE Transactions on Software Engineering* 47, 7 (2019), 1381–1401.

[14] Gerardo Canfora, Massimiliano Di Penta, Rocco Oliveto, and Sebastiano Panichella. 2012. Who is going to mentor newcomers in open source projects?. In *Proc. of the ACM SIGSOFT 20th Int. Symposium on the Foundations of Software Engineering* (Cary, NC, US). ACM, New York, NY, USA, 1–11.

[15] Javier Luis Cánovas Izquierdo and Jordi Cabot. 2022. On the analysis of non-coding roles in open source development. *Empirical Software Engineering* 27, 1 (2022), 1–32.

[16] Andrea Capiluppi, Klaas-Jan Stol, and Cornelia Boldyreff. 2012. Exploring the role of commercial stakeholders in open source software evolution. In *IFIP Int. conf. on Open Source Systems* (Hammamet, Tunisia). Springer, 178–200.

[17] Gemma Catolino, Fabio Palomba, Damian A Tamburri, and Alexander Serebrenik. 2021. Understanding community smells variability: A statistical approach. In *2021 IEEE/ACM 43rd Int. conf. on Software Engineering: Software*

Engineering in Society (Madrid, Spain). IEEE, 77–86.

[18] Gemma Catolino, Fabio Palomba, Damian A Tamburri, Alexander Serebrenik, and Filomena Ferrucci. 2019. Gender diversity and women in software teams: How do they affect community smells?. In *2019 IEEE/ACM 41st Int. conf. on Software Engineering: Software Engineering in Society* (Montreal, Canada). IEEE, 11–20.

[19] Kaylea Champion and Benjamin Mako Hill. 2021. Underproduction: An approach for measuring risk in open source software. In *2021 IEEE Int. conf. on Software Analysis, Evolution and Reengineering*. IEEE, Virtual, 388–399.

[20] Jailton Coelho and Marco Tulio Valente. 2017. Why modern open source projects fail. In *Proc. of the 2017 11th Joint meeting on Foundations of Software Engineering* (Paderborn, Germany). ACM, New York, NY, USA, 186–196.

[21] Jailton Coelho, Marco Tulio Valente, Luciano Milen, and Luciana L. Silva. 2020. Is this GitHub project maintained? Measuring the level of maintenance activity of open-source projects. *Information and Software Technology* 122 (2020), 106274.

[22] Jailton Coelho, Marco Tulio Valente, Luciana L Silva, and André Hora. 2018. Why we engage in FLOSS: Answers from core developers. In *Proceedings of the 11th Int. workshop on Cooperative and Human Aspects of Software Engineering* (Gothenburg Sweden). ACM, New York, NY, USA, 114–121.

[23] Kattiana Constantino, Mauricio Souza, Shurui Zhou, Eduardo Figueiredo, and Christian Kästner. 2021. Perceptions of open-source software developers on collaborations: An interview and survey study. *Journal of Software: Evolution and Process* (2021), e2393.

[24] Eleni Constantinou and Tom Mens. 2017. An empirical comparison of developer retention in the RubyGems and npm software ecosystems. *Innovations in Systems and Software Engineering* 13, 2 (2017), 101–115.

[25] Simone da Silva Amorim, John D McGregor, Eduardo Santana de Almeida, and Christina von Flach G. Chavez. 2018. Educating to achieve healthy open source ecosystems. In *Proc. of the 12th European conf. on Software Architecture: Companion Proc.* (Madrid Spain). ACM, New York, NY, USA, 1–7.

[26] Simone da Silva Amorim, Félix Simas S Neto, John D McGregor, Eduardo Santana de Almeida, and Christina von Flach G. Chavez. 2017. How has the health of software ecosystems been evaluated? A systematic review. In *Proc. of the 31st Brazilian symposium on software engineering* (Fortaleza, CE, Brazil). ACM, New York, NY, USA, 14–23.

[27] Sherae Daniel and Katherine Stewart. 2016. Open source project success: Resource access, flow, and integration. *The Journal of Strategic Information Systems* 25, 3 (2016), 159–176.

[28] Stanislav Dashevskyi, Achim D Brucker, and Fabio Massacci. 2018. A screening test for disclosed vulnerabilities in foss components. *IEEE Transactions on Software Engineering* 45, 10 (2018), 945–966.

[29] Manuel De Stefano, Emanuele Iannone, Fabiano Pecorelli, and Damian Andrew Tamburri. 2022. Impacts of software community patterns on process and product: An empirical study. *Science of Computer Programming* 214 (2022), 102731.

[30] Manuel De Stefano, Fabiano Pecorelli, Damian A Tamburri, Fabio Palomba, and Andrea De Lucia. 2020. Splicing community patterns and smells: A preliminary study. In *Proc. of the IEEE/ACM 42nd Int. conf. on Software Engineering Workshops* (Seoul, Republic of Korea). ACM, New York, NY, USA, 703–710.

[31] Alexandre Decan, Eleni Constantinou, Tom Mens, and Henrique Rocha. 2020. GAP: Forecasting commit activity in git projects. *Journal of Systems and Software* 165 (2020), 110573.

[32] Alexandre Decan, Tom Mens, and Eleni Constantinou. 2018. On the evolution of technical lag in the npm package dependency network. In *2018 IEEE Int. conf. on Software Maintenance and Evolution* (Madrid, Spain). IEEE, 404–414.

[33] Alexandre Decan, Tom Mens, and Eleni Constantinou. 2018. On the impact of security vulnerabilities in the npm package dependency network. In *Proc. of the 15th Int. conf. on Mining Software Repositories* (Gothenburg, Sweden). ACM, New York, NY, USA, 181–191.

[34] Alexandre Decan, Tom Mens, and Philippe Grosjean. 2019. An empirical comparison of dependency network evolution in seven software packaging ecosystems. *Empirical Software Engineering* 24, 1 (2019), 381–416.

[35] Denis Dennehy, Kieran Conboy, Jennifer Ferreira, and Jaganath Babu. 2020. Sustaining open source communities by understanding the influence of discursive manifestations on sentiment. *Information Systems Frontiers* (2020), 1–17.

[36] Remo Eckert and Andreas Mueller. 2017. Sustainability and Diversity of Open Source Software Communities. In *The 12th International conf. on Software Engineering Advances* (Athens, Greece). IARIA, 59.

[37] Evert Eckhardt, Erwin Kaats, Slinger Jansen, and Carina Alves. 2014. The Merits of a Meritocracy in Open Source Software Ecosystems. In *Proc. of the 2014 European conf. on Software Architecture Workshops* (Vienna, Austria). ACM, New York, NY, USA, Article 7, 6 pages.

[38] Vijaya Kumar Eluri, Thomas A Mazzuchi, and Shahram Sarkani. 2021. Predicting long-time contributors for GitHub projects using machine learning. *Information and Software Technology* 138 (2021), 106616.

[39] Vijaya Kumar Eluri, Shahram Sarkani, and Thomas A Mazzuchi. 2019. Open Source Software Survivability Prediction Using Multi Layer Perceptron. In *Proc. of 28th Int. conf. on Software Engineering and Data Engineering*, Vol. 64. EPiC Series in Computing, San Diego, CA, USA, 148–157.

[40] Fabian Fagerholm, Alejandro S Guinea, Jürgen Münch, and Jay Borenstein. 2014. The role of mentoring and project characteristics for onboarding in open source software projects. In *Proc. of the 8th ACM/IEEE Int. Symposium on Empirical Software Engineering and Measurement* (Torino, Italy). ACM, New York, NY, USA, 1–10.

[41] Hongbo Fang, Hemank Lamba, James Herbsleb, and Bogdan Vasilescu. 2022. "This Is Damn Slick!" Estimating the Impact of Tweets on Open Source Project Popularity and New Contributors. In *In 44th Int. conf. on Software Engineering* (Pittsburgh, PA, USA). ACM, New York, NY, USA, 14 pages.

[42] Fabio Ferreira, Luciana Lourdes Silva, and Marco Tulio Valente. 2020. Turnover in Open-Source Projects: The Case of Core Developers. In *Proc. of the 34th Brazilian Symposium on Software Engineering* (Natal, Brazil). ACM, New York, NY, USA, 447–456.

[43] Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The "Shut the f** k up" Phenomenon: Characterizing Incivility in Open Source Code Review Discussions. *Proc. of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–35.

[44] Mívian Ferreira, Marco Tulio Valente, and Kecia Ferreira. 2017. A comparison of three algorithms for computing truck factors. In *2017 IEEE/ACM 25th Int. conf. on Program Comprehension* (Buenos Aires, Argentina). IEEE, 207–217.

[45] Anna Filippova and Hichang Cho. 2016. The effects and antecedents of conflict in free and open source software development. In *Proc. of the 19th ACM conf. on Computer-Supported Cooperative Work & Social Computing* (San Francisco, CA, USA). ACM, New York, NY, USA, 705–716.

[46] Matthieu Foucault, Marc Palyart, Xavier Blanc, Gail C Murphy, and Jean-Rémy Falleri. 2015. Impact of developer turnover on quality in open-source software. In *Proc. of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Bergamo, Italy). ACM, New York, NY, USA, 829–841.

[47] Armstrong Foundjem, Ellis E Eghan, and Bram Adams. 2021. Onboarding vs. Diversity, Productivity, and Quality-Empirical Study of the OpenStack Ecosystem. In *43rd Int. conf. on Software Engineering* (Madrid, Spain). IEEE, 1033–1045.

[48] Oscar Franco-Bedoya, David Ameller, Dolors Costal, and Xavier Franch. 2014. Measuring the quality of open source software ecosystems using queso. In *Int. conf. on Software Technologies* (Druskininkai, Lithuania). Springer, Cham, 39–62.

[49] Oscar Franco-Bedoya, David Ameller, Dolors Costal, and Xavier Franch. 2014. Queso a quality model for open source software ecosystems. In *2014 9th Int. conf. on Software Engineering and Applications* (Vienna, Austria). IEEE, 209–221.

[50] Oscar Franco-Bedoya, David Ameller, Dolors Costal, and Xavier Franch. 2017. Open source software ecosystems: A Systematic mapping. *Information and Software Technology* 91 (2017), 160 – 185.

[51] Felipe Fronchetti, Igor Wiese, Gustavo Pinto, and Igor Steinmacher. 2019. What attracts newcomers to onboard on oss projects? tl; dr: Popularity. In *IFIP Int. conf. on Open Source Systems* (Montreal, QC, Canada). Springer, Cham, 91–103.

[52] Jonas Gamalielsson and Björn Lundell. 2014. Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved? *Journal of Systems and Software* 89 (2014), 128–145.

[53] David Garcia, Marcelo Serrano Zanetti, and Frank Schweitzer. 2013. The Role of Emotions in Contributors Activity: A Case Study on the GENTOO Community. In *2013 Int. conf. on Cloud and Green Computing* (Karlsruhe, Germany). IEEE, 410–417.

[54] Amir Hossein Ghapanchi. 2015. Predicting software future sustainability: A longitudinal perspective. *Information Systems* 49 (2015), 40–51.

[55] Sean Goggins, Kevin Lumbard, and Matt Germonprez. 2021. Open Source Community Health: Analytical Metrics and Their Corresponding Narratives. In *2021 IEEE/ACM 4th Int. Workshop on Software Health in Projects, Ecosystems and Communities* (Madrid, Spain). ACM, New York, NY, USA, 25–33.

[56] Jesus M. Gonzalez-Barahona. 2020. *Characterizing Outdateness with Technical Lag: An Exploratory Study.* ACM, New York, NY, USA, 735–741.

[57] Francielly Grigorio, Daniel Brito, Eudisley Anjos, and Mario Zenha-Rela. 2014. On systems project abandonment: An analysis of complexity during development and evolution of FLOSS systems. In *2014 IEEE 6th Int. conf. on Adaptive Science & Technology* (Ota, Nigeria). IEEE, 1–8.

[58] Mariam Guizani, Amreeta Chatterjee, Bianca Trinkenreich, Mary Evelyn May, Geraldine J. Noa-Guevara, Liam James Russell, Griselda G. Cuevas Zambrano, Daniel Izquierdo-Cortazar, Igor Steinmacher, Marco A. Gerosa, and Anita Sarma. 2021. The Long Road Ahead: Ongoing Challenges in Contributing to Large OSS Organizations and What to Do. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 407 (oct 2021), 30 pages. https://doi.org/10.1145/3479551

[59] Mariam Guizani, Thomas Zimmermann, Anita Sarma, and Denae Ford. 2022. Attracting and Retaining OSS Contributors with a Maintainer Dashboard. In *In 44th Int. conf. on Software Engineering: Software Engineering in Society* (Pittsburgh, PA, USA). ACM, New York, NY, USA, 5 pages.

[60] Hideaki Hata, Taiki Todo, Saya Onoue, and Kenichi Matsumoto. 2015. Characteristics of sustainable oss projects: A theoretical and empirical study. In *2015 IEEE/ACM 8th Int. Workshop on Cooperative and Human Aspects of Software Engineering* (Florence, Italy). IEEE, 15–21.

[61] Zijie Huang, Zhiqing Shao, Guisheng Fan, Jianhua Gao, Ziyi Zhou, Kang Yang, and Xingguang Yang. 2021. Predicting community smells' occurrence on individual developers by sentiments. In *29th Int. conf. on Program Comprehension* (Madrid, Spain). IEEE, 230–241.

[62] Javier Luis Cánovas Izquierdo and Jordi Cabot. 2018. The role of foundations in open source projects. In *Proc. of the 40th Int. conf. on Software Engineering: Software Engineering in Society* (Gothenburg, Sweden). ACM, New York, NY, USA, 3–12.

[63] Abbas Javan Jafari, Diego Elias Costa, Rabe Abdalkareem, Emad Shihab, and Nikolaos Tsantalis. 2021. Dependency smells in Javascript projects. *IEEE Transactions on Software Engineering* 50, 8 (2021), 18 pages.

[64] Slinger Jansen. 2014. Measuring the health of open source software ecosystems: Beyond the scope of project health. *Information and Software Technology* 56, 11 (2014), 1508 – 1519. Special issue on Software Ecosystems.

[65] Oskar Jarczyk, Błażej Gruszka, Szymon Jaroszewicz, Leszek Bukowski, and Adam Wierzbicki. 2014. Github projects. quality analysis of open-source software. In *Int. conf. on Social Informatics* (Barcelona, Spain). Springer, Cham, 80–94.

[66] Oskar Jarczyk, Szymon Jaroszewicz, Adam Wierzbicki, Kamil Pawlak, and Michal Jankowski-Lorek. 2018. Surgical teams on GitHub: Modeling performance of GitHub project development processes. *Information and Software Technology* 100 (2018), 32–46.

[67] Sha Jiang, Jian Cao, and Mukesh Prasad. 2019. The Metrics to Evaluate the Health Status of OSS Projects Based on Factor Analysis. In *CCF conf. on Computer Supported Cooperative Work and Social Computing*. Springer, Cham, 723–737.

[68] Abin Joy, Senthilkumar Thangavelu, and Amalendu Jyotishi. 2018. Performance of GitHub open-source software project: an empirical analysis. In *2018 Second Int. conf. on Advances in Electronics, Computers and Communications* (Bangalore, India). IEEE, 1–6.

[69] Gerta Kapllani, Ilya Khomyakov, Ruzilya Mirgalimova, and Alberto Sillitti. 2020. An Empirical Analysis of the Maintainability Evolution of Open Source Systems. In *IFIP Int. conf. on Open Source Systems* (Innopolis, Russia). Springer, Cham, 78–86.

[70] Jymit Khondhu, Andrea Capiluppi, and Klaas-Jan Stol. 2013. Is it all lost? A study of inactive open source projects. In *IFIP Int. conf. on open source systems* (Koper-Capodistria, Slovenia). Springer, Cham, 61–79.

[71] Apostolos Kritikos and Ioannis Stamelos. 2018. Open Source Software Resilience Framework. In *IFIP Int. conf. on Open Source Systems* (Athens, Greece). Springer, Cham, 39–49.

[72] Stefano Lambiase, Gemma Catolino, Damian A Tamburri, Alexander Serebrenik, Fabio Palomba, and Filomena Ferrucci. 2022. Good Fences Make Good Neighbours? On the Impact of Cultural and Geographical Dispersion on Community Smells. In *In 44th Int. conf. on Software Engineering: Software Engineering in Society* (Pittsburgh, PA, USA). ACM, New York, NY, USA, 12 pages.

[73] Amanda Lee, Jeffrey C Carver, and Amiangshu Bosu. 2017. Understanding the impressions, motivations, and barriers of one time code contributors to FLOSS projects: a survey. In *2017 IEEE/ACM 39th Int. conf. on Software Engineering* (Buenos Aires, Argentina). IEEE, 187–197.

[74] Xiaozhou Li, Sergio Moreschini, Fabiano Pecorelli, and Davide Taibi. 2022. OSSARA: Abandonment Risk Assessment for Embedded Open Source Components. *IEEE Software* 39, 04 (2022), 48–53.

[75] Xiaozhou Li, Sergio Moreschini, Zheying Zhang, and Davide Taibi. 2022. Exploring factors and metrics to select open source software components for integration: An empirical study. *Journal of Systems and Software* 188 (2022), 111255.

[76] Zhifang Liao, Benhong Zhao, Shengzong Liu, Haozhi Jin, Dayu He, Liu Yang, Yan Zhang, and Jinsong Wu. 2019. A prediction model of the project life-span in open source software ecosystem. *Mobile Networks and Applications* 24, 4 (2019), 1382–1391.

[77] Bin Lin, Gregorio Robles, and Alexander Serebrenik. 2017. Developer turnover in global, industrial open source projects: Insights from applying survival analysis. In *2017 IEEE 12th Int. conf. on Global Software Engineering* (Buenos Aires, Argentina). IEEE, 66–75.

[78] Johan Linåker, Efi Papatheocharous, and Thomas Olsson. 2022. Online Suppl. Material. https://doi.org/10.6084/m9.figshare.20137175

[79] Georg JP Link and Debora Jeske. 2017. Understanding organization and open source community relations through the attraction-selection-attrition model. In *Proc. of the 13th Int. Symposium on Open Collaboration* (Galway, Ireland). Association for Computing Machinery, New York, NY, USA, 1–8.

[80] Chao Liu, Dan Yang, Xiaohong Zhang, Haibo Hu, Jed Barson, and Baishakhi Ray. 2018. A recommender system for developer onboarding. In *Proc. of the 40th Int. conf. on Software Engineering: Companion Prc.* (Gothenburg, Sweden). Association for Computing Machinery, New York, NY, USA, 319–320.

[81] Héctor J Macho and Gregorio Robles. 2013. Preliminary lessons from a software evolution analysis of Moodle. In *Proc. of the First Int. conf. on Technological Ecosystem for Enhancing Multiculturality* (Salamanca, Spain). Association for Computing Machinery, New York, NY, USA, 157–161.

[82] Konstantinos Manikas and Klaus Marius Hansen. 2013. Reviewing the health of software ecosystems–a conceptual framework proposal. In *Proc. of the 5th Int. Workshop on Software Ecosystems*. Citeseer, 33–44.

[83] Konstantinos Manikas and Klaus Marius Hansen. 2013. Software ecosystems–a systematic literature review. *Journal of Systems and Software* 86, 5 (2013), 1294–1306.

[84] Konstantinos Manikas and Dimosthenis Kontogiorgos. 2015. Characterizing software activity: The influence of software to ecosystem health. In *Proc. of the 2015 European conf. on Software Architecture Workshops* (Dubrovnik, Cavtat, Croatia). ACM, New York, NY, USA, 1–6.

[85] Junaid Maqsood, Iman Eshraghi, and Syed Sarmad Ali. 2017. Success or failure identification for GitHub's open source projects. In *Proc. of the 2017 Int. conf. on Management Engineering, Software Engineering and Service Sciences* (Wuhan, China). ACM, New York, NY, USA, 145–150.

[86] Jennifer Marlow, Laura Dabbish, and Jim Herbsleb. 2013. Impression formation in online peer production: activity traces and personal profiles in github. In *Proc. of the 2013 conf. on Computer supported cooperative work* (San Antonio, TX, USA). Association for Computing Machinery, New York, NY, USA, 117–128.

[87] Vishal Midha and Prashant Palvia. 2012. Factors affecting the success of Open Source Software. *Journal of Systems and Software* 85, 4 (2012), 895–905.

[88] Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilescu, and Christian Kästner. 2022. "Did You Miss My Comment or What?" Understanding Toxicity in Open Source Discussions. In *In 44th Int. conf. on Software Engineering* (Pittsburgh, PA, USA). ACM, New York, NY, USA, 710–722.

[89] Courtney Miller, David Gray Widder, Christian Kästner, and Bogdan Vasilescu. 2019. Why Do People Give Up FLOSSing? A Study of Contributor Disengagement in Open Source. In *Open Source Systems*, Francis Bordeleau, Alberto Sillitti, Paulo Meirelles, and Valentina Lenarduzzi (Eds.). Springer Int. Publishing, Cham, 116–129.

[90] John Yates Monteith, John D McGregor, and John E Ingram. 2014. Proposed metrics on ecosystem health. In *Proc. of the 2014 ACM Int. workshop on Software-defined ecosystems* (Vancouver, BC, Canada). Association for Computing Machinery, New York, NY, USA, 33–36.

[91] Saya Onoue, Hideaki Hata, and Kenichi Matsumoto. 2014. Software population pyramids: The current and the future of oss development communities. In *Proc. of the 8th ACM/IEEE Int. Symposium on Empirical Software Engineering and Measurement* (Torino, Italy). ACM, New York, NY, USA, 1–4.

[92] Samuel Onyango, Emilie Steenvoorden, Joram Scholten, Slinger Jansen, Peggy Gregory, Philippe Kruchten, et al. 2021. Assessing the Health of the Dark Web: An Analysis of Dark Web Open Source Software Projects. In *Agile Processes in Software Engineering and Extreme Programming–Workshops*, Vol. 426. Springer, 125–134.

[93] Marc Oriol, Oscar Franco-Bedoya, Xavier Franch, and Jordi Marco. 2014. Assessing open source communities' health using Service Oriented Computing concepts. In *2014 IEEE Eighth Int. conf. on Research Challenges in Information Science* (Marrakech, Morocco). IEEE, 1–6.

[94] Khadija Osman and Olga Baysal. 2021. Health is Wealth: Evaluating the Health of the Bitcoin Ecosystem in GitHub. In *2021 IEEE/ACM 4th Int. Workshop on Software Health in Projects, Ecosystems and Communities* (Madrid, Spain). IEEE, 1–8.

[95] Cassandra Overney, Jens Meinicke, Christian Kästner, and Bogdan Vasilescu. 2020. How to Not Get Rich: An Empirical Study of Donations in Open Source. In *Proc. of the ACM/IEEE 42nd Int. conf. on Software Engineering* (Seoul, South Korea). ACM, New York, NY, USA, 1209–1221.

[96] Fabio Palomba and Damian Andrew Tamburri. 2021. Predicting the emergence of community smells using socio-technical metrics: A machine-learning approach. *Journal of Systems and Software* 171 (2021), 110847.

[97] Javier Pérez, Romuald Deshayes, Mathieu Goeminne, and Tom Mens. 2012. Seconda: Software ecosystem analysis dashboard. In *2012 16th European conf. on Software Maintenance and Reengineering* (Szeged, Hungary). IEEE, 527–530.

[98] Etiel Petrinja and Giancarlo Succi. 2012. Two evolution indicators for FOSS projects. In *IFIP Int. conf. on Open Source Systems* (Hammamet, Tunisia). Springer, Cham, 216–232.

[99] Rolf-Helge Pfeiffer. 2021. Identifying critical projects via pagerank and truck factor. In *2021 IEEE/ACM 18th Int. conf. on Mining Software Repositories* (Madrid, Spain). IEEE, 41–45.

[100] Konstantinos Plakidas, Daniel Schall, and Uwe Zdun. 2017. Evolution of the R software ecosystem: Metrics, relationships, and their impact on qualities. *Journal of Systems and Software* 132 (2017), 119–146.

[101] Konstantinos Plakidas, Srdjan Stevanetic, Daniel Schall, Tudor B Ionescu, and Uwe Zdun. 2016. How do software ecosystems evolve? A quantitative assessment of the R ecosystem. In *Proc. of the 20th Int. Systems and Software Product Line conf.* (Beijing, China). Association for Computing Machinery, New York, NY, USA, 89–98.

[102] Gede Artha Azriadi Prana, Abhishek Sharma, Lwin Khin Shar, Darius Foo, Andrew E Santosa, Asankhaya Sharma, and David Lo. 2021. Out of sight, out of mind? How vulnerable dependencies affect open-source projects. *Empirical Software Engineering* 26, 4 (2021), 1–34.

[103] Huilian Sophie Qiu, Yucen Lily Li, Susmita Padala, Anita Sarma, and Bogdan Vasilescu. 2019. The Signals That Potential Contributors Look for When Choosing Open-Source Projects. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 122 (nov 2019), 29 pages.

[104] Huilian Sophie Qiu, Alexander Nolte, Anita Brown, Alexander Serebrenik, and Bogdan Vasilescu. 2019. Going Farther Together: The Impact of Social Capital on Sustained Participation in Open Source. In *2019 IEEE/ACM 41st Int. conf. on Software Engineering* (Montreal, QC, Canada). IEEE, 688–699.

[105] Uzma Raja and Marietta J Tretter. 2012. Defining and evaluating a measure of open source project survivability. *IEEE Transactions on Software Engineering* 38, 1 (2012), 163–174.

[106] Naveen Raman, Minxuan Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. 2020. Stress and Burnout in Open Source: Toward Finding, Understanding, and Mitigating Unhealthy Interactions. In *Proc. of the ACM/IEEE 42nd Int. conf. on Software Engineering: New Ideas and Emerging Results* (Seoul, South Korea). ACM, New York, NY, USA, 57–60.

[107] Mehvish Rashid, Paul M Clarke, and Rory V O'Connor. 2019. A systematic examination of knowledge loss in open source software projects. *Int. Journal of Information Management* 46 (2019), 104–123.

[108] Ayushi Rastogi and Ashish Sureka. 2014. What community contribution pattern says about stability of software project?. In *2014 21st Asia-Pacific Software Engineering conf.* (Jeju, Korea (South)), Vol. 2. IEEE, 31–34.

[109] Leiming Ren, Shimin Shan, Xiujuan Xu, and Yu Liu. 2020. StarIn: An Approach to Predict the Popularity of GitHub Repository. In *Int. conf. of Pioneering Computer Scientists, Engineers and Educators.* Springer, Singapore, 258–273.

[110] Dirk Riehle, Philipp Riemer, Carsten Kolassa, and Michael Schmidt. 2014. Paid vs. volunteer work in open source. In *2014 47th Hawaii Int. conf. on System Sciences* (Honolulu, HI, USA). IEEE, 3286–3295.

[111] Peter C. Rigby, Yue Cai Zhu, Samuel M. Donadelli, and Audris Mockus. 2016. Quantifying and Mitigating Turnover-Induced Knowledge Loss: Case Studies of Chrome and a Project at Avaya. In *2016 IEEE/ACM 38th Int. conf. on Software Engineering* (Austin, TX, USA). IEEE, 1006–1016.

[112] Gregorio Robles and Jesús M González-Barahona. 2012. A comprehensive study of software forks: Dates, reasons and outcomes. In *IFIP Int. conf. on Open Source Systems* (Hammamet, Tunisia). Springer, Cham, 1–14.

[113] Carlos Santos, George Kuk, Fabio Kon, and John Pearson. 2013. The attraction of contributors in free and open source software projects. *The Journal of Strategic Information Systems* 22, 1 (2013), 26–45.

[114] Carlos Denner dos Santos. 2017. Changes in free and open source software licenses: managerial interventions and variations on project attractiveness. *Journal of Internet Services and Applications* 8, 1 (2017), 1–12.

[115] Andreas Schilling, Sven Laumer, and Tim Weitzel. 2012. Who will remain? an evaluation of actual person-job and person-team fit to predict developer retention in floss projects. In *2012 45th Hawaii Int. conf. on System Sciences* (Honolulu, HI, USA). IEEE, 3446–3455.

[116] Thomas Schranz, Christian Schindler, Matthias Müller, and Wolfgang Slany. 2019. Contributors' impact on a FOSS project's quality. In *Proc. of the 2nd ACM SIGSOFT Int. Workshop on Software Qualities and Their Dependencies* (Tallinn, Estonia). ACM, New York, NY, USA, 35–38.

[117] Ravi Sen, Siddhartha S Singh, and Sharad Borle. 2012. Open source software success: Measures and analysis. *Decision Support Systems* 52, 2 (2012), 364–372.

[118] Maha Shaikh and Natalia Levina. 2019. Selecting an open innovation community as an alliance partner: Looking for healthy communities and ecosystems. *Research Policy* 48, 8 (2019), 103766.

[119] Mario Silic and Andrea Back. 2017. Open source software adoption: Lessons from Linux in Munich. *IT Professional* 19, 1 (2017), 42–47.

[120] Diomidis Spinellis. 2019. How to select open source components. *Computer* 52, 12 (2019), 103–106.

[121] Igor Steinmacher, Marco Gerosa, Tayana U Conte, and David F Redmiles. 2019. Overcoming social barriers when contributing to open source software projects. *Computer Supported Cooperative Work (CSCW)* 28, 1 (2019), 247–290.

[122] Igor Steinmacher, Marco Aurelio Graciotto Silva, Marco Aurelio Gerosa, and David F Redmiles. 2015. A systematic literature review on the barriers faced by newcomers to open source software projects. *Information and Software Technology* 59 (2015), 67–85.

[123] Igor Steinmacher, Christoph Treude, and Marco Aurelio Gerosa. 2018. Let me in: Guidelines for the successful onboarding of newcomers to open source projects. *IEEE Software* 36, 4 (2018), 41–49.

[124] Igor Steinmacher, Igor Wiese, Ana Paula Chaves, and Marco Aurélio Gerosa. 2013. Why do newcomers abandon open source software projects?. In *2013 6th Int. Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (San Francisco, CA, USA). IEEE, 25–32.

[125] Igor Steinmacher, Igor Scaliante Wiese, and Marco Aurélio Gerosa. 2012. Recommending mentors to software project newcomers. In *2012 Third Int. Workshop on Recommendation Systems for Software Engineering (RSSE).* IEEE, 63–67.

[126] MM Mahbubul Syeed, Imed Hammouda, and Tarja Systä. 2013. Evolution of open source software projects: A systematic literature review. *J. Softw.* 8, 11 (2013), 2815–2829.

[127] Damian A Tamburri, Fabio Palomba, and Rick Kazman. 2019. Exploring community smells in open-source: An automated approach. *IEEE Transactions on software Engineering* 47, 3 (2019), 630–652.

[128] Damian A Tamburri, Fabio Palomba, Alexander Serebrenik, and Andy Zaidman. 2019. Discovering community patterns in open-source: a systematic approach and its evaluation. *Empirical Software Engineering* 24, 3 (2019), 1369–1417.

[129] Parastou Tourani, Yujuan Jiang, and Bram Adams. 2014. Monitoring sentiment in open source mailing lists: exploratory study on the apache ecosystem. In *24th Annual International conf. on Computer Science and Software Engineering* (Markham, ON, Canada), Vol. 14. IBM, 34–44.

[130] Bianca Trinkenreich, Mariam Guizani, Igor Wiese, Anita Sarma, and Igor Steinmacher. 2020. Hidden figures: Roles and pathways of successful oss contributors. *Proc. of the ACM on human-computer interaction* 4, CSCW2 (2020), 1–22.

[131] Asher Trockman, Shurui Zhou, Christian Kästner, and Bogdan Vasilescu. 2018. Adding Sparkle to Social Coding: An Empirical Study of Repository Badges in the Npm Ecosystem. In *Proc. of the 40th Int. conf. on Software Engineering* (Gothenburg, Sweden). ACM, New York, NY, USA, 511–522.

[132] Marat Valiev, Bogdan Vasilescu, and James Herbsleb. 2018. Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem. In *Proc. of the 2018 26th ACM Joint Meeting on European Software Engineering conf. and Symposium on the Foundations of Software Engineering* (Lake Buena Vista, FL, USA). ACM, New York, NY, USA, 644–655.

[133] Sonny Van Lingen, Adrien Palomba, and Garm Lucassen. 2013. On the software ecosystem health of open source content management systems. In *5th Int. workshop on software ecosystems (iwseco 2013).* Citeseer, 38.

[134] Bogdan Vasilescu, Yue Yu, Huaimin Wang, Premkumar Devanbu, and Vladimir Filkov. 2015. Quality and productivity outcomes relating to continuous integration in GitHub. In *Proc. of the 2015 10th joint meeting on foundations of software engineering* (Bergamo, Italy). ACM, New York, NY, USA, 805–816.

[135] Robert Viseur. 2013. Identifying success factors for the mozilla project. In *IFIP Int. conf. on Open Source Systems* (Koper-Capodistria, Slovenia). Springer, Cham, 45–60.

[136] Paul van Vulpen, Abel Menkveld, and Slinger Jansen. 2017. Health Measurement of Data-Scarce Software Ecosystems: A Case Study of Apple's ResearchKit. In *Int. conf. of Software Business* (Essen, Germany). Springer, Cham, 131–145.

[137] James Walden. 2020. *The Impact of a Major Security Event on an Open Source Project: The Case of OpenSSL.* ACM, New York, NY, USA, 409–419.

[138] Jing Wang. 2012. Survival factors for Free Open Source Software projects: A multi-stage perspective. *European Management Journal* 30, 4 (2012), 352–371.

[139] Zhendong Wang, Yang Feng, Yi Wang, James A Jones, and David Redmiles. 2020. Unveiling elite developers' activities in open source projects. *ACM Transactions on Software Engineering and Methodology* 29, 3 (2020), 1–35.

[140] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proc. of the 18th Int. conf. on evaluation and assessment in software engineering* (London, United Kingdom). ACM, New York, NY, USA, 1–10.

[141] Nebi Yılmaz and Ayça Kolukısa Tarhan. 2022. Quality evaluation models or frameworks for open source software: A systematic literature review. *Journal of Software: Evolution and Process* (2022), e2458.

[142] Likang Yin, Zhuangzhi Chen, Qi Xuan, and Vladimir Filkov. 2021. *Sustainability Forecasting for Apache Incubator Projects.* ACM, New York, NY, USA, 1056–1067.

[143] Yiqing Yu, Alexander Benlian, and Thomas Hess. 2012. An empirical study of volunteer members' perceived turnover in open source software projects. In *2012 45th Hawaii Int. conf. on System Sciences* (Honoulu, HI, USA). IEEE, 3396–3405.

[144] Yang Yue, Yi Wang, and David Redmiles. 2022. Off to a Good Start: Dynamic Contribution Patterns and Technical Success in An OSS Newcomers Early Career. *IEEE Transactions on Software Engineering* (2022), 1–1.

[145] Ahmed Zerouali, Tom Mens, Gregorio Robles, and Jesus M Gonzalez-Barahona. 2019. On the diversity of software package popularity metrics: An empirical study of npm. In *2019 IEEE 26th Int. conf. on software analysis, Evolution and Reengineering* (Hangzhou, China). IEEE, 589–593.

[146] Yuxia Zhang, Hui Liu, Xin Tan, Minghui Zhou, Zhi Jin, and Jiaxin Zhu. 2022. Turnover of Companies in OpenStack: Prevalence and Rationale. *ACM Trans. Softw. Eng. Methodol.* 31, 4, Article 75 (jul 2022), 24 pages.

[147] Jiayuan Zhou, Shaowei Wang, Yasutaka Kamei, Ahmed E Hassan, and Naoyasu Ubayashi. 2022. Studying donations and their expenses in open source projects: a case study of GitHub projects collecting donations through open collectives. *Empirical Software Engineering* 27, 1 (2022), 1–38.

[148] Minghui Zhou and Audris Mockus. 2012. What make long term contributors: Willingness and opportunity in OSS community. In *2012 34th Int. conf. on Software Engineering* (Zurich, Switzerland). IEEE, 518–528.

[149] Minghui Zhou, Audris Mockus, Xiujuan Ma, Lu Zhang, and Hong Mei. 2016. Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects. *ACM Transactions on Software Engineering and Methodology* 25, 2 (2016), 1–29.

[150] Shurui Zhou, Bogdan Vasilescu, and Christian Kästner. 2019. What the Fork: A Study of Inefficient and Efficient Forking Practices in Social Coding. In *Proc.*

of the 2019 27th ACM Joint Meeting on European Software Engineering conf. and Symposium on the Foundations of Software Engineering (Tallinn, Estonia). ACM, New York, NY, USA, 350–361.