On the Self-Governance and Episodic Changes in Apache Incubator Projects: An Empirical Study

Likang Yin
lkyin@ucdavis.edu
University of California, Davis
Davis, California, USA

Xiyu Zhang
xyzzh@ucdavis.edu
University of California, Davis
Davis, California, USA

Vladimir Filkov
vfilkov@ucdavis.edu
University of California, Davis
Davis, California, USA

ABSTRACT

Sustainable Open Source Software (OSS) projects are characterized by the ability to attract new project members and maintain an energetic project community. Building sustainable OSS projects from a nascent state requires effective project governance and socio-technical structure to be interleaved, in a complex and dynamic process. Although individual disciplines have studied each separately, little is known about how governance and software development work together in practice toward sustainability. Prior work has shown that many OSS projects experience large, episodic changes over short periods of time, which can propel them or drag them down. However, sustainable projects typically manage to come out unscathed from such changes, while others do not. The natural questions arise: Can we identify the back-and-forth between governance and socio-technical structure that lead to sustainability following episodic events? And, how about those that do not lead to sustainability?

From a data set of social, technical, and policy digital traces from 262 sustainability-labeled ASF incubator projects, here we employ a large-scale empirical study to characterize episodic changes in socio-technical aspects measured by Change Intervals (CI), governance rules and regulations in a form of Institutional Statements (IS), and the temporal relationships between them. We find that sustainable projects during episodic changes can adapt themselves to institutional statements more efficiently, and that institutional discussions can lead to episodic changes intervals in socio-technical aspects of the projects, and vice versa. In practice, these results can provide timely guidance beyond socio-technical considerations, adding rules and regulations in the mix, toward a unified analytical framework for OSS project sustainability.

I. INTRODUCTION

Sustainable Open Source Software (OSS) projects are characterized by volunteering work, continuous recruitment, and effective governance. However, even OSS projects that are widely used by many, including large companies and national governments, may not attract the attention and resources they need, resulting in unsustainable development and potentially severe consequences downstream [2], [12]. E.g., on December 9th, 2021, the Apache *Log4j* project was reported to have a severe security vulnerability, likely due to being severely under-resourced, affecting countless individuals and organizations. One day later, US government officials assigned the highest severity score for the *Log4j* incident [1].

While not all OSS projects that are unsustainable will have such drastic consequences, many of them that depart from a sustainable trajectory suffer the 'tragedy of the commons' or even get abandoned. This is due to a complex set of circumstances making it challenging to pinpoint and mitigate. On the one hand, software engineering researchers have favored a socio-technical perspective of OSS projects, using email communication and code commits to build socio-technical representations [43], [20], [13]. On the other hand, management science researchers have studied episodic changes in organizations and sustainability for governing the commons, including forests, marine, and fisheries, through the lens of institutional written or unwritten norms, rules, and regulations [44], [3]. By and large, these two perspectives on sustainability have not been fruitfully combined, and very little is known about how effective governance and software development work together in practice toward sustaining OSS projects. But we do have an example that clearly demonstrates how changes in self-governance can change the trajectory of project sustainability.

Motivating Example In Dec 2018, the Apache Software Foundation incubating project OPENWHISK was experiencing a hard time: the number of active developers dropped from 35 to 17 in only a few months. Had such a downturn continued, the project could have gotten abandoned and finally retired from the ASF incubator. Fortunately, a developer started an email thread, noting that there had been insufficient engagement in running the project: '*I've done the release manager role for package release so far, but to me it seems that our release process is being impeded by a lack of engagement from eligible voters on the IPMC mailing list.*' This email thread acted to incentivize others to get positively engaged '*I will reflect this in the quarterly report ... The IPMC is aware of the issue and is currently doing something about that.*' In the months that followed, the project was discussing how to get back on the sustainable trajectory, and finally decided to continue its development by enforcing regulations. '*I would like to see us push out a consolidated next release in the near future (by end of January?). I'd also like to see us attempt*

[1] Log4j incident post:https://nvd.nist.gov/vuln/detail/CVE-2021-44228

to establish a regular cadence of such consolidated releases (perhaps quarterly?)'. During this change event, the number of developers grew from 17 to 28; so did the commits and engagement on the mailing list. Later, the project graduated from the incubator in 2019.

We see in the above that discussions related to norms, rules, and regulations can trigger changes in the software development process, leading to corrections in the OSS project trajectories. We also suspect that OPENWHISK is not the only such example. Prior work has shown that many ASF incubator projects experienced large changes in their socio-technical structure, over short periods of time [53]. Projects with effective governance are more likely to come out unscathed from such large changes, while others do not, implying that governance may be a catalyst to sustainability [52].

Inspired by the above example and prior work, our hypothesis is that concerted institutional discussions can lead to large changes in the underlying socio-technical structure. Symmetrically, changes in the projects' socio-technical structure may require modified rules or institutional governance to be compatible with the new structure. To validate our hypothesis, in this work we chose a set of 262 Apache Software Foundation (ASF) incubator projects. The ASF incubator is a well-known pioneer and champion for open source. It hosts hundreds of OSS projects, striving to nurture sustainable communities in each project through ASF-wide mechanisms, including a set of institutional policies and governance. When a project exits the incubator, each project is evaluated and labeled as graduated (sustainable) or retired (unsustainable) by ASF committees. Such an extrinsic labeling is essential to understanding sustainability. Just as importantly, the openness and completeness of the ASF mailing lists (ASF's tenet is 'If it didn't happen on the mailing list, it didn't happen'), makes the ASF incubator a key resource for studying OSS sustainability from both the socio-technical, software engineering and the institutional governance perspectives.

Starting from a data set of social, technical, and policy digital traces from 262 sustainability-labeled ASF incubator projects, and guided by related social and organizational theories, here we sought to study the more specific hypothesis:

> Sustainable projects can process and translate self-governance rules and policies into socio-technical changes, and vice versa, more effectively than unsustainable projects.

To that end, we employ a large-scale empirical study to characterize sustainable and unsustainable projects by matching episodic changes in their socio-technical structure to evidence of institutional discussions. We operationalize this by matching episodic time-series events, i.e., Change Intervals (CI) in the socio-technical structure, to sentence-level institutional discussions, i.e., Institutional Statements (IS), as well as the temporal relationships between them. We develop a framework for simultaneous, socio-technical and institutional, analysis of OSS projects, with a view to describing and understanding a process affected by both, namely, projects gaining self-sustainability and self-government and eventually graduating from the ASF incubator. Our findings are as follows:

- We can effectively identify episodic Change Intervals (CI) in the socio-technical structure, and they tend to be temporally co-located with Institutional Statements (IS);
- CIs have effects at both individual-level and project-level, and such effects vary across both agents and projects with different levels of sustainability;
- During episodic changes, sustainable (i.e., ASF incubator graduated) projects can convert institutional rules into practice more efficiently than other projects.

To the best of our knowledge, this work is among the first to attempt to study the structural changes in OSS projects and their self-governance under a unified analytical framework. We are hopeful that refining this convergent approach, of socio-technical and institutional analyses, will lead to new ways of thinking about and analyzing emergent properties in modern software engineering such as OSS sustainability.

II. Background and Theoretical Framework

This section introduces the background and social theories pertinent to OSS governance and sustainability.

A. *Theory of Governing the Commons*

A major portion of Ostrom's Nobel Prize-winning work [31] investigated how individuals collaborate and create self-governing institutions in natural resource settings [47], e.g., water [6], marine [21], and forest [18]. However, in practice, individuals who cannot be easily excluded from the use of shared natural resources often have little incentive to contribute to the production or maintenance of these resources [30], [37]. This refers to the 'Tragedy of the Commons' [33], and these individuals are often referred to as free-riders in natural resource commons settings [30], [37].

In the OSS context, OSS code is clearly an inexhaustible resource to users: it can be copied over and over. However, there are exhaustible (limited) resources in OSS commons, e.g., developer's efforts. But there is also maintenance that is regularly needed, e.g., on defects, technical debt, etc. The combination of limited developer effort available and the need to keep technical debt low produces situations similar to the tragedy of the commons, since developers find building new features more rewarding than performing code maintenance or fixing bugs in OSS projects. In that sense, OSS free-riders, would be those favoring feature development over fixing bugs.

Over the course of a lifetime, Ostrom demonstrated through hard work in the development of self-governing institutions that communities can avert such tragedy [31]. This was accomplished primarily through the introduction and evolution of the Institutional Analysis and Development (IAD) framework [32]. We and others before us have realized Ostrom's formalism is appropriate for analyzing OSS commons [19], [23], [40], where exogenous factors are: the socio-technical context as community attributes, ASF's and project-specific regulations as the rules-in-use, and the biophysical conditions correspond to software artifacts being developed. The action arena consists of OSS contributors and action situations. Of course, sometimes the concepts fit very well, and other times, as in our answer about free-riders above, the concept matching is more distant, and that's where our current and future work lies: in extending IAD and Ostrom's rules to OSS ecosystems. However, even if over-appropriation may not be a problem for OSS, the tragedy of the commons can still happen in the OSS context, and OSS sustainability lies at the core of the solution. Such tragedy arises when there are free-riders who do not provide sufficient work on development and maintenance while taking the spot, therefore, the project cannot achieve the functionality and use intended, and thereafter becomes abandoned [39].

B. Organizational Change Theory

One can gain a more comprehensive understanding of the nuances of organizational change through the interaction between different perspectives, because every theoretical perspective provides a partial account of a complex phenomenon [50]. Here we present three main pillars of organizational change theory.

Episodic Change Organizational changes are viewed as episodic changes when they occur infrequently, discontinuously, and intentionally [14]. Organizations tend to undergo episodic change during periods of divergence when they move away from their equilibrium condition [50]. Developing divergence results from a growing misalignment between an inertial deep structure and perceived environmental demands.

Agents During organizational change, influencers who are committed to change are viewed as change agents. With their charisma and fortitude, such agents motivate and lead their teams by engaging them in the change process. These kinds of leadership pedigrees can be found in two types [26]: The first type of agent uses power as a means of rewarding and sanctioning their staff. The second type of agent has the trust of their staff, and in these cases charismatic agents can successfully influence others to follow their commands.

Resistance In effect, episodic changes in an organization can cause employees' resistance in the workplace [48]. Piderit et. al. [34] claim that people tend to stay unchanged in their workspace, as their primary responsibility might be the welfare of their families. Therefore any organizational change that is going to impact that reality is going to encounter some kind of resistance if the employees are not involved in the change process. As such, successful organizational adaptation is increasingly reliant on generating employee support and enthusiasm for proposed changes, rather than merely overcoming resistance [34].

C. Socio-Technical System Theory

OSS projects, and the socio-technical side of software engineering in general, have dominated the analysis for a long time through organizational and socio-technical perspectives [5], [25], [41], [9]. Social-technical systems (STS) consist of two main components: the social part, where users continuously create and share knowledge by engaging in various kinds of interactions with one another [11], and the technical part, where they rely on technical hardware to accomplish collective tasks [49], [7], [38]. STS is typically referred to when examining how a technical system is able to provide efficient and reliable interaction between individuals. In addition, it examines how the social subsystem is affected by interactions and therefore influences the performance of the technical system [22], [17]. One might also describe STS as an intermediary entity that transfers institutional influence to individuals, combining the views of engineers and social scientists [36]. Xuan et al. [51] propose a method to measure the interleaving effects between email communications and code commits in OSS projects, and they find that bursts in communications before and after code commits are essential for effective software development. From the STS perspective, the ASF community is a unique system that has both outside influence regulations from the ASF board and members and inside structure managed or self-governed committees.

III. Research Questions

In the previous section, we reported that a variety of scholars have utilized a socio-technical approach to analyze complex collective behaviors in OSS projects. We also described how institutional analysis is useful in understanding institutional governance under the context of digital commons (i.e., OSS projects). Moreover, prior work has shown that episodic changes to the socio-technical network can be indicators of changes in project sustainability [53]. As a first step to linking such changes to their antecedents in institutional discussions, we focus on the methodology for identifying large changes in socio-technical features, over time. We ask:

RQ_1: Are there episodic changes in the socio-technical structure of projects during their incubation? Likewise, can we identify discussions related to policy and governance?

As predicted by organizational change theory, episodic changes are associated with agents and will be reflected in some form of resistance (e.g., negative developers' engagement, responsiveness, and sentiment). We ask:

RQ_2: Is there significant resistance evident in policy/governance-related discussions associated with episodic change? How do they differ between sustainable projects and others?

Per institutional analysis theory, strategies, norms, and rules can affect the socio-technical structure of projects. In addition, institutional governance and organizational structure must work hand-in-hand to make viable socio-technical systems. Ill-designed institutional arrangements can introduce inefficiencies into the system, which may amplify non-standard behavior and structure. In a sustainable system, an ill-formed organizational structure may induce new rules to adjust and improve such structure, improving efficiencies in the systems. Therefore, such influential links from institutional design to the organizational structure can be, in fact, bi-directional. Most such changes will motivate some and demotivate other developers, which will manifest as variable sentiment in their communication.

Thus, we hypothesize that the feedback loop, if any, between institutional governance and organizational structure can be quantitatively measured and associated with overall sentiment. We ask:

RQ$_3$: What are the associations between episodic change direction (i.e., up-turns and down-turns) in socio-technical structure and the sentiment in IS-related discussions?

In the following section, we introduce the methodologies approaching the above three research questions.

IV. Data and Methods

In this work, we leverage a previously published data set [54] consisting of hundreds of Apache Software Foundation Incubator projects. ASF Incubator (ASFI) aims to help projects become self-sustaining and eventually join ASF [42]. The incubation outcome is two-fold: One is graduation indicating that the project has a self-sustainable community to move it forward, otherwise the project is retired.

On the data end, in addition to the previously published data set, we gather complementary time-stamped trace data of commits and emails using PERCEVAL [15]. To reduce the noise caused by outliers in email data, we removed bots' automated emails by applying regular expressions to email titles and content. Similarly, for commit data, we use *GitHub Linguist* and identify 731 collective programming language and markup file extensions to remove non-coding commits (e.g., committed to files with extension .json, .jpg, .png, etc.). Our final data contains 262 projects, among them, 205 are graduated projects, and 57 projects are retired. In total, we collect 1,548,807 email records from 42,191 unique emails contributors, and 359,297 commit records from 5,931 unique ASF committers [2].

A. Constructing Socio-technical Networks

Studies of complex systems, such as OSS projects, have largely relied on network science approaches [4], [46], [24]. As socio-technical networks can contain both information about the components (i.e., the nodes) and the interactions between the components (i.e., the edges), we use them here as abstraction anchors. In this work, the socio-technical networks consist of two types of networks [27]: social networks, which are extracted from their email communications, and technical networks, based on commits to source files. At each month in incubation, for each project, we form social networks (weighted directed graphs) from the communications between developers as follows: developer A has a directed edge to developer B only if B has replied to A's post on the mailing lists during that month. The edge weight represents the frequency of communication between two developers. The technical weighted bipartite graph is formed in a similar way. We include an undirected edge between developer A and a source file F if developer A has committed to file F during that month. Each edge is weighted according to the frequency at which it is committed to the source file. We use the *Python* NETWORKX package for the network implementation.

B. Identifying Institutional Statements (IS)

Through the lens of open-access email discussions among ASF committers, ASF mentors, and other types of contributors, we can then capture their institutional designs in the form of ISs.

Definition of Institutional Statement (IS). We refer to a sentence-level institutional discussion as an *Institutional Statements* (IS). For example, on 24 Feb 2017, an ASF incubator project Airflow sent out an email containing institutional statements "*Next steps: 1) will start the voting process at the IPMC mailing list. ... So, we might end up with changes to stable. ... 2) Only after the positive voting on the IPMC and finalisation I will rebrand the RC to Release.*" In short, norms, rules, and strategies are outlined as prescriptions and constraints that mobilize and organize actors for collective action in a form of institutional statements. To extract IS from the email corpus, we leverage previous work on institutional analysis [52]. As there is no ground truth for institutional statements to train the IS classifier, they first hand-annotated a small subset of the data for IS as follows. Using a random subset of 313 email threads from incubator project lists, two coders classified each sentence in them as either 'IS' or 'Not IS' according to whether it came from an institutional statement or not, which results in 6,805 labeled sentences (i.e., 'IS' or 'Not IS'), and there were 273 of them labeled as IS [3]. They combined the email exchange data set to fine-tune a BERT-based classifier [10], for automatic detection of ISs. In the end, given the fact that ISs are rare (there are only about 5% emails contain ISs), and the task naturally is challenging, the classifier achieved a precision score of 0.667, recall score of 0.681, and F1 score of 0.674 on classifying institutional statements, showing the classifier is able to extract ISs from developer email exchanges.

C. Identifying Change Intervals (CI)

Organizational changes are categorized as episodic changes when they occur infrequently, discontinuously, and intentionally. When studying the dynamics of socio-technical systems, prominent changes in socio-technical network variables that intuitively mark critical events are particularly relevant.

Definition of Change Intervals (CI). We refer to the time periods (in months) during which these episodic changes occur as *Change Intervals* (CI). We use the Cumulative Sum (CuSum) algorithm to detect these change intervals with the package DETECTA. CuSum algorithm is a widely used method for monitoring abrupt changes in time-series data [16]. A typical form of CuSum algorithm is to calculate the cumulative sums in positive and negative directions along an axis of the data and mark an alarm point when reaching some threshold c. DETECTA can extract the increasing/decreasing change interval containing an alarm point. It also uses a drift parameter,

[2]Our code is available at Zenodo: https://doi.org/10.5281/zenodo.6526833

[3]Coding manual: https://doi.org/10.5281/zenodo.7042616

denoted by d, to penalize a long, flat drift. Our parameter selection procedure on c and d respects the diverse properties of different projects. Specifically, for a socio-technical variable, each project gets its unique pair of c and d, calculated based on the project's fluctuation level. In the procedure, we first calculated the pairwise differences ($|x_i - x_j|$, where x_i is the data point at time $t = i$) in the data for each project. Then, within each project, we took the ratio between the mean μ^* of the largest 20% of the pairwise differences and the overall mean μ, denoted by $p = \frac{\mu^*}{\mu}$. This ratio p from all projects formed a distribution P which provides a comprehensive view of the extent to which large changes in these projects outstrip project averages. From distribution P, we select a value for the base parameter p_0, which is used to generate unique c and d for each project, with the equations: $c = p_0\mu$ and $d = 0.1c$. To restrict the $Type\ I$ error rate, we conservatively set $p_0 = P_{0.75}$, where $P_{0.75}$ is the 75^{th} percentile of P. We demonstrate an exemplary CI in Figure 1.

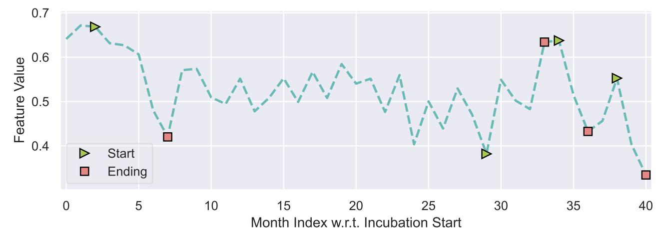

Fig. 1. An illustration of a change interval (CI).

D. *Variables of Interest*

§ Socio-technical Recent work shows that the network modeling is exhibiting high predictive power for OSS success and sustainability [25], [1], [45]. Our socio-technical network variables are pulled from a recent study on forecasting the sustainability of OSS projects [53], All metrics are aggregated on a monthly basis for each project. In total, we have ten socio-technical network variables. The first five are in the social network: (1) number of nodes `s_num_nodes` indicates the unique active developers in social networks; (2) average clustering coefficient `s_avg_clustering_coef` describes the linkage of a node to its neighbors, measured by the closed triplets divided by all triplets; (3) number of components `s_num_component` is the total number of disconnected components in the social networks; (4) weighted mean degree `s_weighted_mean_degree` represents the mean degree of the social networks; (5) graph density `s_graph_density` measures the density of the network, calculated as the number of existing edges divided by the number of all possible edges; And the other five variables in the technical network are: (6) number of developer nodes `t_num_dev_nodes` is the number of unique developers in the technical networks; (7) number of file nodes: `t_num_file_nodes` is the number of unique coding files; (8) number of developers per file node `t_num_dev_per_file` measures the degree of collaborative behaviors; (9) number of files per developer node `t_num_file_per_dev` describes the degree of multitasking behaviors; (10) graph density `t_graph_density` represents the density of the network, calculated as the number of existing edges divided by the number of all possible edges in the technical networks.

§ Institutional In addition to institutional discussions, we define the variables indicating agents and resistance during episodic changes. First, the change agents are categorized into three classes below: (a) Mentors, who give mentorship to projects and help them grow and build their community toward sustainability. Mentors may intervene in the projects if the projects are not progressing well. (b) Committers, who are the major component of the workers for building the artifact. In a project, an individual becomes a committer until they make their first actual code commit to the code base. (c) Contributors, who present the largest population in the community. Contributors are the individuals who are neither mentors nor the committer, i.e., they do not contribute code changes nor mentor the projects. However, contributors are essential to OSS sustainability. They can be helpful in the sense of non-code contributions, e.g., writing documentation, testing, and providing feedback. As predicted by the institutional change theory, during episodic changes, certain resistance may occur in the organization. Resistance measures are calculated on a monthly basis. (a) Responsiveness. Responsiveness is the first level of resistance by slowing down their work pace. It is measured by the average delay time (in days) when agents reply to previous IS-related emails. (b) Engagement. Engagement is the second level of resistance by not participating in certain discussions. It is calculated by the average number of emails the agent engaged in. (c) Negativity. Negativity is the third level of resistance and it carries the opposite information to the discussion. It is measured by calculating the number of negative emails over all emails. Since the negative content is much zero-inflated, we only look at the top 20% negative periods in our data, and use the Mann-Whitney U test to test the shift in means. We summarize the above variables in Table I

E. *Sentiment Detection*

To detect the sentiment in institutional discussions, we use the state-of-the-art NLP model from package PYSENTIMIENTO to extract opinions from texts [35], which first came out in 2021. The base model is BERTweet, a RoBERTa model [28] trained on tweets, which is designed to handle sentiment and emotion analysis tasks in social discussions [29]. In our setting, the task is to extract sentiment from the discussions about institutional statements, we find that such a task is suitable to use BERTweet since the institutional statements are similar to open discussions in tweets. Previous work, Senti4SD [8], aims to address sentences like 'kill this process' under code-mixed software engineering context. Despite the fact that Senti4SD is trained within software engineering context. We believe that the BERT-based PYSENTIMIENTO model is, arguably, more suitable for our task, as discussions

TABLE I
DEFINITION OF VARIABLES.

Type	Variable	Definition
Social-technical	s_num_nodes	The number of unique active developers in the social network.
	s_num_component	The total number of disconnected components in the social network.
	s_graph_density	Density of the social network as the number of existing edges divided by all possible edges
	s_avg_clustering_coef	The average clustering coefficient as the ratio of closed triplets divided by all triplets.
	s_weighted_mean_degree	The mean degree of the social network.
	t_num_dev_nodes	The number of unique developers in the technical network.
	t_num_file_nodes	The number of unique source code files in the technical network.
	t_num_dev_per_file	The degree of developer collaborations as the number of developers per file node.
	t_num_file_per_dev	The degree of multitasking as the number of files per developer node.
	t_graph_density	Density of the social network as the number of existing edges divided by all possible edges
Institutional:Agents	Mentor	Person who mentors projects and helps them grow a sustainable community.
	Committer	Person who commits and reviews code changes.
	Contributor	Person who contributes through non-coding activities, e.g., providing user feedback.
Institutional:Resistance	Responsiveness	The average delay time (in days) for agents to reply to IS-related emails.
	Engagement	The average number of emails that an agent engages in.
	Negativity	The number of negative emails over all emails for each agent.

about institutional are more akin to social discussions rather than technical discussions with code snippets. To reduce the noise in the sentiment data, we only keep informative replies that are classified as either positive or negative.

V. RESULTS

In this section, we study the change in OSS projects by adopting a unified framework, from the institutional analysis and socio-technical system perspectives.

A. ***RQ***$_1$*: Are there episodic changes in socio-technical structure during project incubation? Likewise, can we identify institutional discussions?*

As described in Sect. IV, we used a change interval detection model to help us identify events in the socio-technical structure of ASF incubator projects. Here, we present descriptive statistics about the change intervals we detected.

In Figure 2, we show the total number of change intervals across all socio-technical structures per project. We aggregate the projects by their sustainability status accordingly (i.e., graduated or retired). We find that graduated projects tend to have a higher value in the total change intervals, suggesting that the development of graduated projects is more fluctuating than the retired projects. The total length of the change intervals is averaged by each socio-technical variable per project. We find that, for most of the change intervals, they tend to occupy 20% of the project incubation time, while the change intervals in retired projects have around 20% to 30% for their incubation time.

On the other hand, we are also interested in the stats from our IS classifier. As we can see in Figure 3, the number of monthly IS of graduated projects is more than retired ones'. Moreover, the number of IS from retired projects has a decreasing trend from month 2 to month 6 while for graduated projects it is more stable from month 1 to 12.

We have shown descriptive statistics about the detected change intervals, in the following sections we will show how to use the CI and IS more practically.

> **RQ**$_1$ **Summary:** We showed that most change intervals tend to occupy about 20% of the projects' incubation time. We demonstrated that graduated (i.e., sustainable) projects have more institutional statements and shorter change intervals than retired (i.e., unsustainable) projects.

B. ***RQ***$_2$*: Are there significant resistances in IS-related discussions associated with episodic change? How do such temporal patterns differ across graduated and retired projects?*

As predicted by institutional change theory, during an episodic change in the project's organizational structure, the OSS volunteers may incur an extra workload. In addition, different agents have varying levels of importance during episodic change, and in reverse, they are influenced by episodic change differently, i.e. resistance to change. In this section, to study the association between institutional statements and episodic change intervals, we dive deeper into three aspects at the project level: responsiveness, engagement, and negativity with respect to the change interval. Together, they represent three different levels of observable resistance in projects. Here, we only measure such change regarding IS-related discussions, that is, only the email exchanges containing IS are included. The three resistance measures are defined in Sect. IV-D.

We show the response delay over three periods, i.e., one month before the episodic change (pre-CI), during the episodic change (within-CI), and one month after the episodic change (post-CI) in Figure 4. All values are normalized as per month and aggregated by groups. The result of the Mann-Whitney U test suggests that there exists a significant increase in the means of response delay (in days) from pre-CI to within-CI,

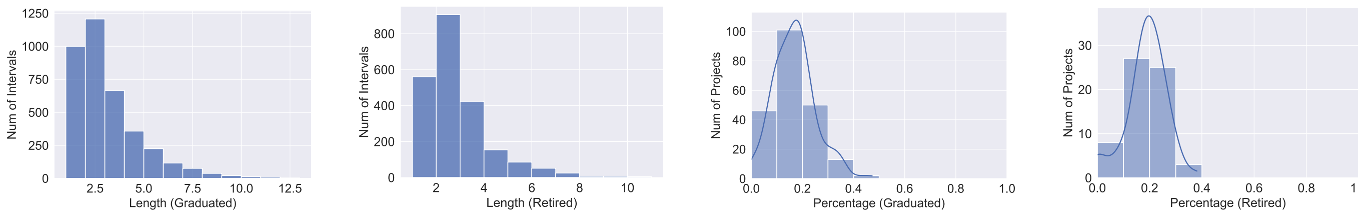

Fig. 2. The distribution of the change interval duration (in months) and percentage w.r.t. projects' incubation time averaged across all socio-technical variables.

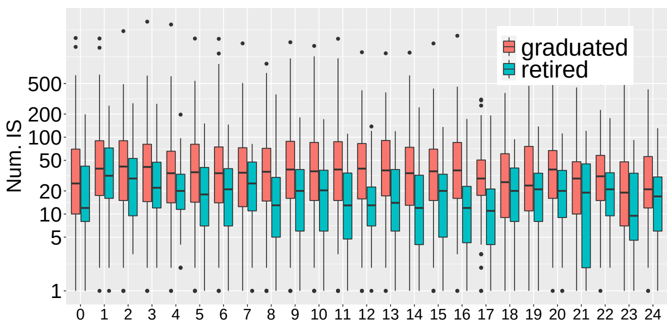

Fig. 3. The monthly num. of institutional statements by sustainability status.

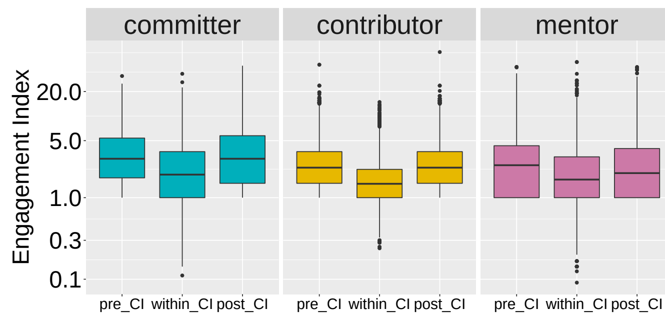

Fig. 5. Engagement over pre-, within-, and post-CI periods.

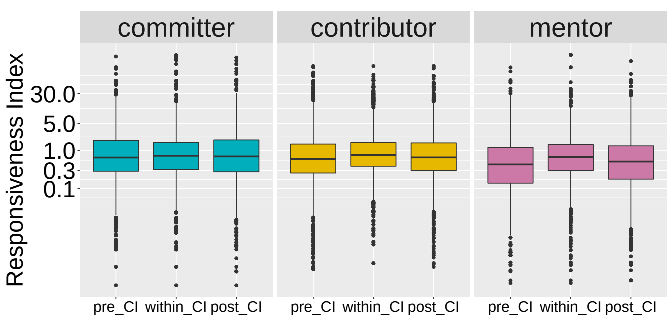

Fig. 4. Responsiveness over pre-, within-, post-CI periods.

and from within-CI to post-CI periods, for both contributors and mentors, with p-value <.001. We find that the shift in committers' responsiveness is much lower and insignificant from pre-CI to within-CI period than contributors and mentors, suggesting the episodic change has a more significant effect on contributors and mentors, rather than committers, in terms of responsiveness. A possible reason for this phenomenon is the fact that committers are required to maintain a consistent commitment to development even during episodic changes.

Next, we look at the second level of resistance, engagement. As a reminder, the engagement is quantified by the number of IS-related emails for each type of agent engaged per month. The engagement indicates the level of participation in institutional issues (e.g., regulation, rules, and norms) within the projects. Unlike the responsiveness index, agents can refuse to engage in certain IS-related issues during an episodic change period, and such reactions, if significant, should be observable via statistical analysis. As shown in Figure 5, the Mann-Whitney U test suggests that the engagement of all three types of agents in IS-related discussions is significantly reduced during the episodic change with p-values <.001. Combined with the results of responsiveness in Figure 4, we consider one possible reason for this to be that committers are not slowing down because they are more focused on specific types of issues, allowing them to remain an almost constant response rate in those. On the other hand, mentors are the delegates of regulations in projects but not the stack-holder, therefore, they only need to attend to certain IS-related issues if projects are progressing well.

For the last attribute, we look at the negativity. The negativity index is calculated for all types of agents by the number of negative emails over all emails on a monthly basis. We measure how much agents oppose IS-related issues before, during, and after episodic changes. As shown in Figure 6, we measure such changes in the pre-, within-, and post-change interval periods. We find that three agents exhibit three different patterns in the negativity measure. Committers have a significant negativity increase between pre-CI and within-CI period, as suggested by the Mann-Whitney U test with p-value <.001. Moreover, we find that the mentors tend to be, in general, more negative than both committers and contributors regarding IS-related issues. In addition, contributors have steadily increasing shifts in negativity and much more outliers, while mentors have a significant increase in negativity only from within-CI to the post-CI period. There are the following possible reasons for this phenomenon: (1) Committers are the stakeholders, and they are the ones that will follow the regulations and norms, imposing episodic changes will shift them to be more negative on discussing IS-related issues. (2)

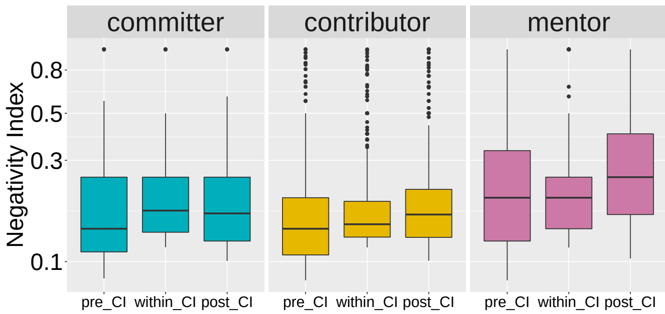

Fig. 6. Negativity over pre-, within-, and post-CI periods.

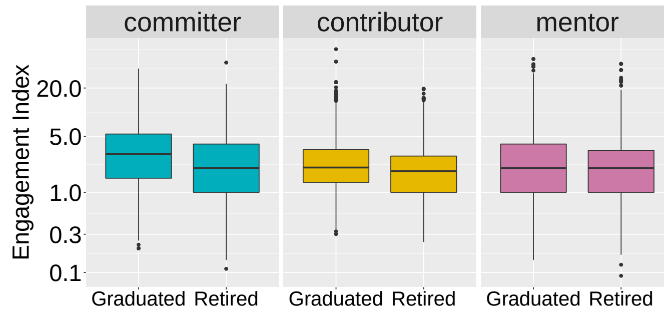

Fig. 8. Engagement over graduated and retired projects.

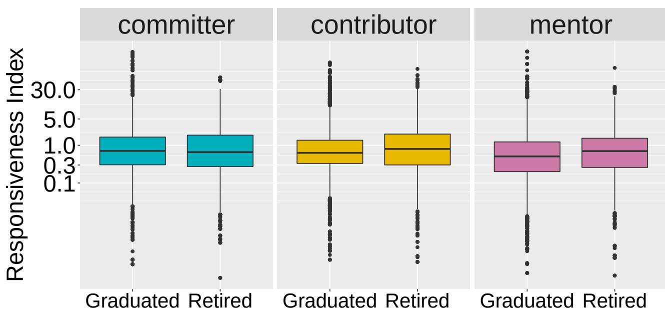

Fig. 7. Responsiveness over graduated and retired projects.

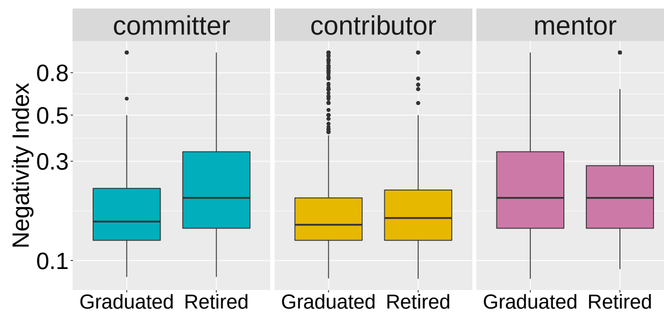

Fig. 9. Negativity over graduated and retired projects.

Contributors are more of a customer of OSS rather than a producer, they are the less conservative people in the projects, and they are more likely to complain about regulations and rules they do not want to have, therefore, forming outliers. (3) Mentors' criticism is followed by the episodic changes, while during the episodic changes they tend to be more supportive of the team and help the community go through the changes.

In addition to time intervals around episodic changes, we also study the influence of episodic changes vary across projects with different sustainability levels, i.e., graduated and retired projects. As shown in Figure 7, for responsiveness, we find that there are significant differences in means between graduated projects and retired projects in contributors and mentors, but not committers, with p-value <.001. The possible reason for this phenomenon is the fact that ASF committers in both graduated and retired projects are almost equally responsive to issues. In addition, graduated projects have a more diverse community that can respond to issues more promptly, while retired projects may have a limited size of community members who work from the same time zone.

Then, we show that there exists a significant difference in committers' and contributors' engagement between graduated and retired projects (p-value <.001), as shown in Figure 8. The committers, as expected, have a higher level of engagement (the means in graduated and retired projects are 4.47 and 3.58, respectively), suggesting that developers in graduated projects are more interested in discussions regarding community building on institutional governance.

As shown in Figure 9, we find that both committers and contributors in retired projects are significantly more negative to IS-related discussions as compared to graduated projects. It suggests that in the retired projects, there is no capacity for people to move their projects forward regarding building up regulations and rules, i.e., committers in retired projects are 'burn-out' when they keep up the same response rate in graduated projects.

> **RQ$_2$ Summary:** We find that, around change intervals, there exist certain resistances, and they vary across different types of agents. E.g., contributors and mentors become more negative while committers are less negative after episodic changes while. In addition, in retired (unsustainable) projects, contributors and mentors exhibit lower response rates; committers and contributors are much less engaged and more negative.

C. Case study: Association between episodic change and institutional statements

To communicate concretely how the institutional and sociotechnical dimensions interact within ASF ecosystem, we showcase two diverse instances of their mutual interrelationship.

§ Case A: From IS to CI. *Calcite* (former *Optiq*) is a project migrated to Apache Incubator from another platform. From the initial months since *Calcite* was accepted into Apache Incubator, we detected a series of IS preceding a sharp increasing CI in the average social network clustering coefficient (`s_avg_clustering_coef`), which reflects a growth in the connectivity between people in the project's social network. After closely examining the IS, we found that this IS→CI interrelationship revealed meaningful events.

In June 2014, the second month since their migration, the developers started arranging online meetings and calling for connections as an effort to adapt to the new community and its workflow. Below are examples of such IS from two of the developers, Dev_1 and Dev_2:

Dev_1 *"I'd like to hear from people across the community, users as well as developers, people who have contributed a two-line patch six months ago as well as people who check-in daily. ... I propose that we have an online meeting to introduce ourselves, using Google hangouts. I'd especially like to meet people who would like to get involved by writing documentation, blog posts, and testing."*

Dev_1 *"We can discuss at the online meeting. I volunteer to write a draft report on Monday."*

Dev_2 *"Before we start opening issues there, I would like to discuss here if you want to import the GitHub issues into JIRA."*

Shortly after, in July 2014, another IS shows that Dev_1 proposed for their first and subsequent releases on Apache:

Dev_1 *"Optiq has been releasing regularly, but it is important that Optiq soon makes an initial release under the Apache process."*

Following these IS, from July to October 2014, the project's social network connectivity skyrocketed. An explanation of this CI is that the project was under its migration process to the Apache platform. Instead of gradually building up a community, a migrated project like *Calcite* transferred its established community to the new platform within a relatively small amount of time. Moreover, the initiation of regular releasing also triggered interactions and enhanced this surge.

§ Case B: From CI to IS. From March to May 2016, the *Quickstep* project experienced sharp growth in its number of developers (t_num_dev_nodes) as indicated by our change detection results. Immediately after this CI, in June 2016, we detected a cluster of IS. This CI→IS interrelationship helped us locate a notable phase of the project when developers started to raise concerns in response to their rapidly growing community.

A developer first expressed their concern about the lack of on-list discussion in opposition to the frequent development activities with the following IS: *"There seems to be quite a lot of work happening on project, but I can't figure out where the design discussions and decisions are being made. ... Where are design discussions happening? Does the team have a weekly or daily meeting? I ask because if discussions are happening off-list, and in particular if decisions are being made off-list, the project is not attractive to outsiders..."*.

In a separate discussion thread, the developer also suggested that a clear criterion and process regarding committer election was needed and favorable to attracting and retaining committers with the following IS: *"You, as a PMC, should decide what are the criteria & process for making someone a committer... Electing committers is a consequence of a successful strategy for growing community, and helps further that growth."*

D. RQ3: What are the associations between episodic change direction and the sentiment to IS-related discussions?

In previous RQs, we conducted exploratory and qualitative studies of project discussion and dynamics around episodic changes. In this section, we triangulate those with quantitative studies, to understand the pre-cursor to episodic changes in terms of the sentiment in IS-related discussions, and vice versa. Such an approach enables us to understand how episodic changes in organizational structure features can introduce sentiment shifts in discussions on rules and regulations, and vice versa.

We present the occurrences of a 4-way combination of CI and IS in the table for both graduated and retired projects, as shown in Table II. Each row represents a feature in the socio-technical system, while each column stands for a specific case in the association from Change Interval (CI) to Institutional Statements (IS), and vice versa. For example, the column entitled CI^+IS^+ indicates that there is an increasing trend in a feature that serves as a precursor to positive IS discussions in the following month. CI can be either increasing (+) or decreasing (-), and the sentiment is either positive (+) or negative (-). For measuring the sentiment of IS discussions in respective months, we aggregate the sentiment across all IS discussions and get the majority sentiment.

For graduated projects, as shown in Table II, we find that the number of nodes (s_num_nodes) in social networks accounts for most types of occurrences of CI and IS (168), suggesting that graduated projects are experiencing episodic changes with respect to the total number of active developers in social networks. We continue to use the tables in practice. The tables can also help us understand what is the precursor to positive/negative discussion in IS-related discussions, i.e., agreement/disagreement to rules and regulations. E.g., to find out the episodic change of which socio-technical feature is more likely to be followed by positive discussions, we can focus on the first 4 columns, e.g., CI^+IS^+, CI^+IS^-, CI^-IS^+, and CI^-IS^-, and then calculate the ratio of the cases having IS+ as the outcome, shown as the values in parentheses. In the case of graduated projects, it shows that a positive sentiment followed by an increasing episodic change in s_num_nodes has the highest ratio of 64%, suggesting that we can make people more positive about regulations and rules by recruiting more new-comers on the mailing list. One reason for this result is the fact that the newcomers are not the stakeholders, and they are more open to rules and norms.

As another use case of this table, we can ask the following question: What are the effects that sentiment has on the change interval of the number of unique committers (t_num_dev_nodes)? From Table II, for retired projects, we find that the ratios of having positive sentiment and negative sentiment followed by an increasing trend of t_num_dev_nodes are 29% to 7%, respectively. It suggests that, in retired projects, odds of positive sentiment on IS-related discussions to attract new committers are more than four times than negative sentiment.

TABLE II
THE TEMPORAL ASSOCIATION BETWEEN EPISODIC CHANGE INTERVAL (CI) AND SENTIMENT IN THE INSTITUTION STATEMENT (IS). THE VALUE SHOWS THE OCCURRENCES OF RESPECTIVE CASES (RATIO IN PARENTHESES). THE UNDERLINED CELLS INDICATE THE VARIABLES THAT HAVE THE MOST FREQUENT PATTERN.

Status	Feature	$CI^+ \rightarrow IS^+$	$CI^+ \rightarrow IS^-$	$CI^- \rightarrow IS^+$	$CI^- \rightarrow IS^-$	$IS^+ \rightarrow CI^+$	$IS^+ \rightarrow CI^-$	$IS^- \rightarrow CI^+$	$IS^- \rightarrow CI^-$
Graduated	s_avg_clustering_coef	101 (0.48)	13 (0.06)	78 (0.37)	20 (0.09)	67 (0.39)	85 (0.49)	8 (0.05)	12 (0.07)
	s_graph_density	41 (0.25)	11 (0.07)	91 (0.55)	23 (0.14)	35 (0.32)	57 (0.52)	9 (0.08)	8 (0.07)
	s_num_component	50 (0.49)	5 (0.05)	43 (0.42)	4 (0.04)	50 (0.49)	45 (0.44)	3 (0.03)	5 (0.05)
	s_num_nodes	167 (0.64)	26 (0.1)	57 (0.22)	11 (0.04)	127 (0.6)	63 (0.3)	12 (0.06)	11 (0.05)
	s_weighted_mean_degree	94 (0.51)	13 (0.07)	67 (0.36)	12 (0.06)	80 (0.49)	68 (0.42)	7 (0.04)	8 (0.05)
	t_graph_density	83 (0.45)	7 (0.04)	84 (0.46)	9 (0.05)	59 (0.45)	51 (0.39)	11 (0.08)	9 (0.07)
	t_num_dev_nodes	138 (0.62)	20 (0.09)	55 (0.25)	8 (0.04)	92 (0.56)	58 (0.35)	9 (0.05)	6 (0.04)
	t_num_dev_per_file	104 (0.61)	14 (0.08)	47 (0.27)	6 (0.04)	49 (0.46)	43 (0.4)	11 (0.1)	4 (0.04)
	t_num_file_nodes	73 (0.51)	10 (0.07)	48 (0.34)	12 (0.08)	60 (0.47)	52 (0.41)	10 (0.08)	5 (0.04)
	t_num_file_per_dev	62 (0.44)	8 (0.06)	62 (0.44)	10 (0.07)	49 (0.4)	54 (0.45)	11 (0.09)	7 (0.06)
Retired	s_avg_clustering_coef	38 (0.54)	8 (0.11)	19 (0.27)	5 (0.07)	30 (0.39)	33 (0.43)	4 (0.05)	9 (0.12)
	s_graph_density	17 (0.35)	7 (0.15)	20 (0.42)	4 (0.08)	25 (0.45)	21 (0.38)	8 (0.15)	1 (0.02)
	s_num_component	18 (0.56)	2 (0.06)	10 (0.31)	2 (0.06)	15 (0.39)	16 (0.42)	4 (0.11)	3 (0.08)
	s_num_nodes	35 (0.39)	12 (0.13)	32 (0.36)	10 (0.11)	32 (0.37)	40 (0.46)	4 (0.05)	11 (0.13)
	s_weighted_mean_degree	41 (0.39)	8 (0.08)	43 (0.41)	12 (0.12)	28 (0.35)	43 (0.53)	5 (0.06)	5 (0.06)
	t_graph_density	34 (0.42)	2 (0.03)	38 (0.48)	6 (0.07)	25 (0.3)	43 (0.51)	7 (0.08)	9 (0.11)
	t_num_dev_nodes	49 (0.44)	7 (0.06)	51 (0.46)	4 (0.04)	26 (0.29)	51 (0.57)	6 (0.07)	7 (0.08)
	t_num_dev_per_file	35 (0.43)	2 (0.02)	35 (0.43)	9 (0.11)	12 (0.17)	47 (0.65)	4 (0.06)	9 (0.12)
	t_num_file_nodes	35 (0.39)	4 (0.04)	46 (0.52)	4 (0.04)	29 (0.34)	47 (0.55)	8 (0.09)	1 (0.01)
	t_num_file_per_dev	33 (0.39)	2 (0.02)	41 (0.48)	9 (0.11)	26 (0.33)	45 (0.58)	5 (0.06)	2 (0.03)

To account for the bias in the distribution of episodic change type and sentiment in IS-related discussions, we normalize all values, feature-wise, by taking the ratio of the occurrences over all events, e.g., normalized $CI^+IS^+ = CI^+IS^+ / (CI^+IS^+ + CI^+IS^- + CI^-IS^+ + CI^-IS^-)$. And then we aggregate all features into four 2×2 matrices, two for graduated projects and two for retired projects, as shown in Figure 10. To compare patterns between graduated projects and retired projects, we first look at the first column of the left two matrices (i.e., CI→IS). We find that graduated projects are more likely to have positive discussions after an increasing episodic change than retired projects (49.9% to 6.9% than 43.2% to 7.2%), suggesting that retired projects do not establish a feedback cycle for project progress, or maybe even worse, that they do not realize the progress they made. Then we attend to the second matrix for graduated and the last matrix for retired projects (i.e., all from IS→CI), and we find that, for both graduated and retired projects, positive institutional discussions often serve as the precursor event to episodic changes (88.0% for graduated projects and 84.8% for retired projects).

RQ3 Summary: In graduated projects, the control features for outcome sentiment to IS lay mostly in social networks, while for retired projects, they are more evenly distributed in both social and technical networks. For both graduated and retired projects, positive IS-related discussions are more likely to be followed by a episodic change. While the graduated projects are more likely to have positive discussions after an increase in CI than retired projects.

VI. TAKEAWAYS FOR PRACTITIONERS

In this section, we distill our findings into some practical takeaways and suggestions.

Like in other self-governed institutions, norms, rules, and regulations in OSS projects generate, moderate and direct actions. Concerted governance efforts can result in episodic changes in the socio-technical structure, achieving feedback between effective self-governance and sustainability. Generally, we found more and longer episodic change intervals in graduated projects than in retired ASF projects, potentially explained by the observation that institutional discussions often trigger episodic changes in the project's socio-technical structure. However, institutional discussions are mostly lacking during such changes, so our first takeaway is that just like participation in the technical aspects, developers should be encouraged and even brought into project institutional discussions. We found that episodic changes are associated with temporary developer disengagement from the project, and that the negativity of mentors increases significantly after episodic changes. New episodic changes can be distracting to a team and may bring more management efforts to the project. Setting manageable expectations for the team ahead of time can limit feelings of frustration arising out of lengthy discussions. Thus, being more positive than negative may help keep change intervals shorter. On the other hand, we found that projects that graduate are much less negative toward changes compared to projects that retire. Fostering positive discussions may help the project adapt to changes and become more sustainable. As another takeaway, perhaps projects can benefit from timing episodic changes, to the extent possible, to occur during periods of low cross-team interactions/collaboration. That can potentially ease the cost of upcoming episodic changes.

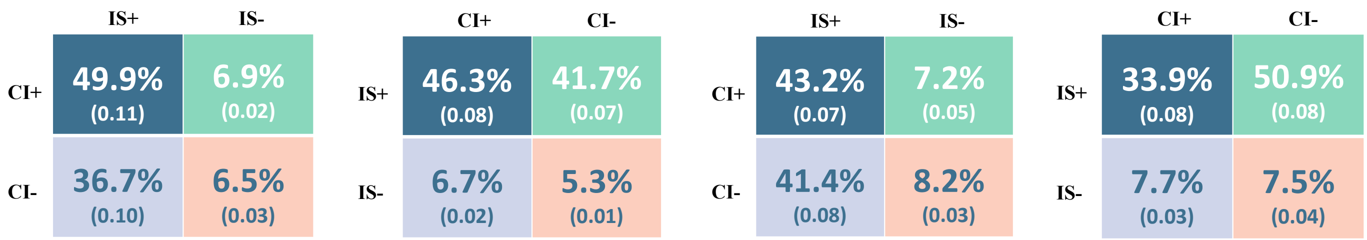

Fig. 10. The aggregated ratio of two-way transmission effects from CI to IS, and vice versa. From left to right, they stand for graduated (CI→IS), graduated (IS→CI), retired (CI→IS), retired (IS→CI). We show the standard deviation in parentheses.

VII. Threats and Conclusions

§ Threats Generalizing our findings beyond ASF, or even beyond the ASF Incubator projects carries potential risks, for example, not every OSS foundation has a mentor program like ASF's. The risk could be reduced by expanding the dataset beyond the ASF incubator, e.g., to include additional projects from other OSS incubators. ASF mailing lists are the only channel we consider, therefore, developers may communicate through in-person meetings, webpage documentation, and private emails. ASF mandates the use of public mailing lists for most project discussions, which causes an especially low risk of omitting institutional or socio-technical information. Annotations of institutional statements could be biased by individual annotators. However, given that the annotators were adequately trained with given reference materials, which lowers the possibility of bias. ASF developers may use different aliases or emails making it difficult to identify distinct developers, while ASF's regulations on using *apache.org* official email addresses and our de-aliasing process reduce such risks [4].

§ Conclusions Practitioners may be able to improve their individual practices, organizational management, and institutional structure by understanding why open-source projects cannot meet the expectations of nonprofit foundations. Additionally, it is important to consider how institutional design and socio-technical aspects relate to OSS to understand its potential sustainability. Through the artifacts they create, we demonstrated that socio-technical network features can capture the episodic change in the organizational structure of developers' collaboration and communication. Through the unified view of socio-technical network features and institutional analysis, we leverage the unique attributes of Apache Software Foundation's Incubator projects to extend the modeling of OSS project sustainability, by analyzing a longitudinal dataset consisting of vast text and log corpora, as well as extrinsic labels for sustainable and unsustainable.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This material is based upon work supported by the National Science Foundation under GCR grant no. 2020751 and no. 2020900.

[4] ASF committer emails: https://infra.apache.org/committer-email.html

References

[1] Bao, L., Xia, X., Lo, D., and Murphy, G. C. A large scale study of long-time contributor prediction for github projects. *IEEE Transactions on Software Engineering 47*, 6 (2019), 1277–1298.

[2] Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., and Venters, C. C. Sustainability design and software: The karlskrona manifesto. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering* (2015), vol. 2, IEEE, pp. 467–476.

[3] Berkowitz, A. D. An overview of the social norms approach. *Changing the culture of college drinking: A socially situated health communication campaign 1* (2005), 193–214.

[4] Bird, C., Nagappan, N., Gall, H., Murphy, B., and Devanbu, P. Putting it all together: Using socio-technical networks to predict failures. In *2009 20th International Symposium on Software Reliability Engineering* (2009), IEEE, pp. 109–119.

[5] Bird, C., Pattison, D., D'Souza, R., Filkov, V., and Devanbu, P. Latent social structure in open source projects. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering* (2008), pp. 24–35.

[6] Blomquist, W., et al. *Dividing the waters: governing groundwater in Southern California.* ICS Press Institute for Contemporary Studies, 1992.

[7] Bock, T., Hunsen, C., Joblin, M., and Apel, S. Synchronous development in open-source projects: A higher-level perspective. *Automated Software Engineering 29*, 1 (2022), 1–53.

[8] Calefato, F., Lanubile, F., Maiorano, F., and Novielli, N. Sentiment polarity detection for software development. *Empirical Software Engineering 23*, 3 (2018), 1352–1382.

[9] Cataldo, M., Herbsleb, J. D., and Carley, K. M. Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement* (2008), pp. 2–11.

[10] Cohan, A., Beltagy, I., King, D., Dalvi, B., and Weld, D. S. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (Hong Kong, China, 2019), Association for Computing Machinery, p. 3693–3699.

[11] Crowston, K., and Howison, J. The social structure of free and open source software development. *First Monday 10*, 2 (2005).

[12] Crowston, K., Wei, K., Howison, J., and Wiggins, A. Free/libre open-source software development: What we know and what we do not know. *ACM Comput. Surv. 44*, 2 (mar 2008).

[13] Dawood, K. A., Sharif, K. Y., Zaidan, A., Abd Ghani, A. A., Zulzalil, H. B., and Zaidan, B. Mapping and analysis of open source software (oss) usability for sustainable oss product. *IEEE Access 7* (2019), 65913–65933.

[14] Dawson, P. Reflections: On time, temporality and change in organizations. *Journal of Change Management 14*, 3 (2014), 285–308.

[15] Dueñas, S., Cosentino, V., Robles, G., and Gonzalez-Barahona, J. M. Perceval: software project data at your will. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings* (2018), ACM, pp. 1–4.

[16] Ellaway, P. Cumulative sum technique and its application to the analysis of peristimulus time histograms. *Electroencephalography and clinical neurophysiology 45*, 2 (1978), 302–304.
[17] Fischer, G., and Herrmann, T. Socio-technical systems: a meta-design perspective. *International Journal of Sociotechnology and Knowledge Development (IJSKD) 3*, 1 (2011), 1–33.
[18] Fleischman, F., Loken, B., Garcia-Lopez, G., and Villamayor-Tomas, S. Evaluating the utility of common-pool resource theory for understanding forest governance and outcomes in indonesia between 1965 and 2012. *International Journal of the Commons 8*, 2 (2014).
[19] Frischmann, B., Madison, M., and Strandburg, K. *Governing Knowledge Commons*. Oxford University Press, 2014.
[20] Ghapanchi, A. H. Predicting software future sustainability: A longitudinal perspective. *Information Systems 49* (2015), 40–51.
[21] Gruby, R. L., and Basurto, X. Multi-level governance for large marine commons: politics and polycentricity in palau's protected area network. *Environmental science & policy 33* (2013), 260–272.
[22] Herrmann, T., Hoffmann, M., Kunau, G., and Loser, K.-U. A modelling method for the development of groupware applications as socio-technical systems. *Behaviour & Information Technology 23*, 2 (2004), 119–135.
[23] Hess, C., and Ostrom, E. *Understanding knowledge as a commons: From theory to practice*. JSTOR, 2007.
[24] Howison, J., Wiggins, A., and Crowston, K. Validity issues in the use of social network analysis with digital trace data. *Journal of the Association for Information Systems 12*, 12 (2011), 2.
[25] Joblin, M., and Apel, S. How do successful and failed projects differ? a socio-technical analysis. *ACM Trans. Softw. Eng. Methodol.* (dec 2021).
[26] Krysinski, P. R., and Reed, D. B. Organizational change and change leadership. *Journal of Leadership Studies 1*, 2 (1994), 65–72.
[27] Kwan, I., Schroter, A., and Damian, D. Does socio-technical congruence have an effect on software build success? a study of coordination in a software project. *IEEE Transactions on Software Engineering 37*, 3 (2011), 307–324.
[28] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[29] Nguyen, D. Q., Vu, T., and Nguyen, A. T. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200* (2020).
[30] Olson, M. The logic of collective action [1965]. *Contemporary Sociological Theory 124* (2012).
[31] Ostrom, E. *Governing the commons: The evolution of institutions for collective action*. Cambridge university press, 1990.
[32] Ostrom, E. *Understanding institutional diversity*. Princeton university press, 2009.
[33] Ostrom, E. *Tragedy of the Commons*. Palgrave Macmillan UK, London, 2016, pp. 1–5.
[34] Piderit, S. K. Rethinking resistance and recognizing ambivalence: A multidimensional view of attitudes toward an organizational change. *Academy of management review 25*, 4 (2000), 783–794.
[35] Pérez, J. M., Giudici, J. C., and Luque, F. pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks, 2021.
[36] Ropohl, G. Philosophy of socio-technical systems. *Techné: Research in Philosophy and Technology 4*, 3 (1999), 186–194.
[37] Sandler, T. *Global collective action*. Cambridge University Press, 2004.
[38] Sarker, F., Vasilescu, B., Blincoe, K., and Filkov, V. Socio-technical work-rate increase associates with changes in work patterns in online projects. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (2019), IEEE, pp. 936–947.
[39] Schweik, C. M., and English, R. Tragedy of the foss commons? investigating the institutional designs of free/libre and open source software projects. *First Monday* (2007).
[40] Schweik, C. M., and English, R. C. *Internet success: a study of open-source software commons*. MIT Press, 2012.
[41] Sharma, A., Prana, G. A. A., Sawhney, A., Nagappan, N., and Lo, D. Analyzing offline social engagements: An empirical study of meetup events related to software development. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)* (2022), IEEE, pp. 1122–1133.
[42] Stănciulescu, S., Yin, L., and Filkov, V. Code, quality, and process metrics in graduated and retired asfi projects. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (2022), pp. 495–506.
[43] Subramaniam, C., Sen, R., and Nelson, M. L. Determinants of open source software project success: A longitudinal study. *Decision Support Systems 46*, 2 (2009), 576–585.
[44] Sunder, S. Minding our manners: Accounting as social norms. *The British accounting review 37*, 4 (2005), 367–387.
[45] Surian, D., Lo, D., and Lim, E.-P. Mining collaboration patterns from a large developer network. In *2010 17th Working Conference on Reverse Engineering* (2010), IEEE, pp. 269–273.
[46] Surian, D., Tian, Y., Lo, D., Cheng, H., and Lim, E.-P. Predicting project outcome leveraging socio-technical network patterns. In *2013 17th European Conference on Software Maintenance and Reengineering* (2013), IEEE, pp. 47–56.
[47] Tainter, J. A. Social complexity and sustainability. *ecological complexity 3*, 2 (2006), 91–103.
[48] Thomas, R., and Hardy, C. Reframing resistance to organizational change. *Scandinavian Journal of Management 27*, 3 (2011), 322–331.
[49] Trist, E. *The evolution of socio-technical systems: A conceptual framework and an action research program*. Ontario Ministry of Labour, 1981.
[50] Van de Ven, A. H., and Poole, M. S. Explaining development and change in organizations. *Academy of management review 20*, 3 (1995), 510–540.
[51] Xuan, Q., Gharehyazie, M., Devanbu, P. T., and Filkov, V. Measuring the effect of social communications on individual working rhythms: A case study of open source software. In *2012 International Conference on Social Informatics* (2012), IEEE, pp. 78–85.
[52] Yin, L., Chakraborti, M., Yan, Y., Schweik, C., Frey, S., and Filkov, V. Open source software sustainability: Combining institutional analysis and socio-technical networks. *Proc. ACM Hum.-Comput. Interact. 6*, CSCW2 (nov 2022).
[53] Yin, L., Chen, Z., Xuan, Q., and Filkov, V. Sustainability forecasting for apache incubator projects. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (New York, NY, USA, 2021), Association for Computing Machinery, p. 1056–1067.
[54] Yin, L., Zhang, Z., Xuan, Q., and Filkov, V. Apache software foundation incubator project sustainability dataset. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)* (2021), IEEE, pp. 595–599.