Do We Run How We Say We Run? Formalization and Practice of Governance in OSS Communities

Mahasweta Chakraborti
University of California Davis
Department of Communication
Davis, CA, USA
mchakraborti@ucdavis.edu

Curtis Atkisson
University of Massachusetts Amherst
School of Public Policy
Amherst, MA, USA
atkissoncj@gmail.com

Ștefan Stănciulescu
University of California Davis
Department of Computer Science
Davis, CA, USA
contact@stefanstanciulescu.com

Vladimir Filkov
University of California Davis
Department of Computer Science
Davis, CA, USA
filkov@cs.ucdavis.edu

Seth Frey
University of California Davis
Department of Communication
Davis, CA, USA
sethfrey@ucdavis.edu

ABSTRACT

Open Source Software (OSS) communities often resist regulation typical of traditional organizations. Yet formal governance systems are being increasingly adopted among communities, particularly through non-profit project-sponsoring foundations. Our study looks at the Apache Software Foundation Incubator program and 208 of the projects it has supported. We assemble a scalable, semantic pipeline to discover and analyze the governance behavior of projects from their mailing lists. We then investigate the relationship of such behavior to what the formal policies prescribe, through their own governance priorities and how their members internalize them. Our findings indicate that a greater amount of policy over a governed topic doesn't elicit more governed activity on that topic, but does predict greater internalization by community members. Moreover, alignment of community operations with foundation governance, be it dedicating their governance focus or adopting policy along topics seeing greater policy-making, has limited association with project outcomes.

CCS CONCEPTS

• **Human-centered computing → Computer supported cooperative work**; **Empirical studies in collaborative and social computing**; *Social engineering (social sciences)*; **Open source software**; *Empirical studies in HCI*.

KEYWORDS

Open Source Software, Peer Production, Online Communities, Collective Action, OSS Governance

ACM Reference Format:
Mahasweta Chakraborti, Curtis Atkisson, Ștefan Stănciulescu, Vladimir Filkov, and Seth Frey. 2024. Do We Run How We Say We Run? Formalization and Practice of Governance in OSS Communities. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 26 pages. https://doi.org/10.1145/3613904.3641980

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3641980

1 INTRODUCTION

An exemplary instance of online peer production [3], Open Source Software (OSS) has emerged as a multi-billion dollar informal industry supporting major contemporary tech enterprises, academia, and scientific research and development. OSS projects have often observed some degree of overarching coordination and governance [46, 76] to manage their products and mentor their developers. Well-laid-out, written *formal* policies are often used to steer and synchronize community operations, thus minimizing the costs of coordination and management [8, 26]. Alongside such formal rules, communities have simultaneously observed other *informal* practical norms to structure activities, assign responsibilities, utilize project resources, and ensure sustained development [16, 31, 35, 46, 52, 57, 59, 103].

Market pressures and recent trends are driving standardization in the OSS ecosystem, including greater adoption of centralized governance models. Over the past three decades, the increasing stakes of OSS have paved the way for several non-profit OSS foundations, which support hundreds of projects and implement systematic governance over their communities. These organizations serve OSS through mentoring, much-needed infrastructure [32] (servers, centralized storage, etc.), legal aid around licensing [67], and well-maintained technical support [62]. Foundations have brought OSS mainstream visibility, attracting even larger numbers of contributors and financial support [57].

Consequently, community governance of mentored projects is a product of the foundation's formal policies, the community's own informal norms and practices, and any interactions between those two sources of institutional structure. Hence, even among projects under the same foundation, their decisions, actions, and ensuing interactions may reflect varied degrees of involvement with the centralized governance, as they may prefer to manage their community in their own fashion. To researchers, OSS foundations and their projects are an ideal sandbox for observing governance dynamics across online institutions.

Non-profit OSS foundations are steadily rising in popularity, with one survey finding 101 active organizations that host over 1,600 OSS projects as of 2018 [38, 39]. With mentored projects generally showing higher survival rates over independent communities [80, 104], they are being increasingly viewed as a model to raise thriving projects producing usable, compliant software. Yet, OSS governance is not without its quirks and challenges [16, 46, 83]. Indeed, there have been instances where formal governance has produced little impact or has actually limited community flexibility and autonomy [41, 67, 83].

Notable among OSS foundations is the Apache Software Foundation (ASF). The Apache Software Foundation Incubator (ASFI) was founded by the Apache Software Foundation (ASF) in 2002 in part to propagate their approach to OSS governance and has mentored over 300 projects ('podlings') since. Several non-profits require interested projects to undergo initiation through an incubation program and learn the ways and requirements of the foundation. ASFI also evaluates projects for performance and overall organizational fit throughout their incubation before accepting ('graduating') them for continued support or 'retiring' them from the foundation. ASFI is famously committed to fostering self-governance and sustained operations in its communities. This raises a fundamental question: what is the relationship of each project's emergent governance structure to the formal policies representing governance across the ASFI?

Our study focuses on community-level governance among mentored projects and how they relate to the common policies imposed by their foundation. We leverage developer conversations from ASFI's public mailing lists (which, by policy, contain all project business). Compared to traditional approaches like surveys, interviews, or other forms of qualitative inference, retrieving behavioral measures from trace data is faster, convenient for replication across foundations while offering more granular, real-time insight. We assess each project's governance efforts and resulting operational structuring through the routinized *governed activities* they perform. Governed activities are measured through recurring [11, 47] operations discussed among email exchanges. Next, we evaluate *policy internalization* of projects, i.e., the extent to which ASFI formal policies and their elements structure community governance and frame their governed activities. Internalization is measured as the semantic similarity between governed activities and formal policies within topics covered by ASFI policies. We analyze how the extent of community governance efforts and policy internalization relate to ASFI's extent of regulation (number of rules) across different governance topics. Finally, we empirically investigate how community governance and the extent of their formal policy internalization together explain its outcome in the ASF Incubator. Our contributions and findings are as follows:

(1) We demonstrate a scalable approach, based on semi supervised learning, to understand governance across peer production communities through both its formal specification and lived instantiation.
(2) A foundation-level analysis of ASFI projects shows that the extent of policy regulation — the number of rules structuring different governance topics — is not mirrored in practice through the extent of governed activity. Yet governed activity tends to internalize formalization to a greater extent in domains where there are more policies, as indicated through the semantic similarity between developer discussions and formal policy. Therefore, while there is greater formal policy internalization in governance topics that are laid out with more rules, these topics with more rules do not necessarily elicit more governed activity among communities.
(3) When it comes to sustaining the community and efficient development towards graduation, more governed activity and greater policy internalization of topics highly regulated/ prioritized in formal policies had little association with the odds of success. All in all, formalized policies in OSS communities may not accurately reflect their underlying patterns of "lived" governance structure or community success.

2 REVIEW: OSS MANAGEMENT

Open Source governance includes all organizational structures and coordination mechanisms that regulate community interactions as well as product development. Prior work has extensively explored OSS community governance in terms of decision-making [31, 40, 104], assignment of tasks [16, 57], managing developer roles and access [35, 59], mentorship [82], code quality, review, and contribution [46, 89], etc.

Community governance has been treated as an expansive, multi-level system of mutually interactive socio-technical networks [25, 41]. Meanwhile, Schweik et al. studied OSS projects at scale on SourceForge and found governance structures to be generally informal and lean, with increased sophistication and formal rules as communities grew [78]. Similar findings are also echoed in O'Mahoney's work on the Debian Linux community's evolving governance [63]. Geiger et al. interpreted the complexity of OSS governance through the increasing labor involved from maintainers as projects scale [28]. Community-level analysis of Apache Incubator projects also found that more successful projects showed greater adoption and use of definitive rules and norms [104]. Heckmann et al.'s investigation of decision-making processes further found that in well-performing projects, developers and users participated more proactively in steering the course of the project [31].

Leadership is a crucial aspect of OSS governance, where developers with greater technical initiative, development prowess, and effective communication strategies generally emerge to fill administrative roles [33]. Analysis of decision episodes in communities found administrators to be critical drivers during the initial phases of a project [31]. Meanwhile, Atkisson specifically examined individual mentors of the Apache Incubator and found a significant correlation between who mentored a project and its odds of graduation [1]. Investigation of communities on SourceForge found that while a sizeable fraction (around 15-20%) of successful projects comprised a stable community with dedicated users, the rest showed rapid growth and were often led by a 'benevolent dictator' [79, 80].

Prior work has explored the challenges of OSS moderation. Qiu et al. designed interactive dashboards to help project maintainers track multiple metrics of community health and foster inclusive participation [71]. Attempts towards greater inclusiveness by enforcing community codes of conduct (CoCs) have often received limited engagement or been perceived as distractions from core development priorities [48]. Several studies have focused on interactions within foundation-led communities. A qualitative cost-benefit analysis of Apache Incubator policies found that the implementation efforts and payoffs are evenly balanced between projects and the ASFI [82]. The implications of congruence/dissonance become particularly salient when it concerns software licensing. The rigor of the licensing requirements, including ASF's rights over individual contributions, has often seen varied reception and interpretation among OSS developers [67]. Sun's introduction of changes in the Netbeans licensing scheme threatened the collapse of the very project [41]. Stringent terms set by corporations supporting gated OSS communities often turn away sincere contributors. Moreover, such formalization also restricts usage of the product, thus hindering developer engagement and community health [83].

While prior work has either focused on foundations or community dynamics, a limited number [104] have empirically treated the mutual interactions between practical operations and formal policies unraveling in real-time [66]. Moreover, they have generally focused on a particular aspect of governance, such as licensing, through case studies of a select number of projects. We attempt to capture the multifacetedness of OSS governance (including but not limited to licensing, trademarks, documentation, committees, voting, etc.) and study hundreds of mentored projects. Motivated by collective action theory and behavior in communities of practice, we proceed to investigate the governance behavior of OSS communities around formalization.

3 THEORETICAL BACKGROUND

Institutional Polycentrism in Communities of Practice

In collective action research, institutions are defined as "… prescriptions that humans use to organize all forms of repetitive and structured interactions …" [65]. Therefore, for a collectively maintained resource such as an OSS community, governance includes all formal and informal rules for management and production, along with the mechanisms for such policy design, reform, and implementation. [54, 80].

OSS projects are essentially online communities of practice [7, 45, 64], where coordinated operations are expressed as routines. Routines comprise structured activities across project communities. These arise from informal rules, beliefs, cognitive scripts, and habitual conventions and translate into 'repeated patterns of actions' across appropriate settings [11, 47]. Routines include management, standard operating procedures, e.g., workflows, or experiential strategies encoded into everyday activities and associated interactions [17, 47, 61]. Community routines may not be only technical and may also emerge to coordinate developers through informal norms and social control [35, 52, 59]. For example, developers use their particular routines for managing and deploying builds, incorporating patches, testing, prioritizing issues, etc. Similarly, they also perform managerial routines for setting up committees, organizing conferences, and ratifying releases.

OSS governance lies on the spectrum between purely self-interest-driven individual action ("the invisible hand") and intentional governance [18]. With the growing role of foundations and other support organizations, OSS governance is increasingly polycentric. Polycentricity refers to a condition where there are overlapping interests between multiple centers of authority, often manifested as multiple levels [42, 53, 54]. Formalized incubators like the ASFI establish systematically planned, overarching policies to coordinate and promote community engagement across all the diverse projects, which also have their own autonomy to establish and execute policy within the framework set by the foundation [7, 47]. ASFI policy introduces specific roles, offices, leadership responsibilities, processes, and protocols for articulating the many facets of managing an OSS project in the ASF style. Within the framework provided by the ASFI and its parent ASF, project-level developer routines reflect how project-level governance builds from ASFI guidance to interpret ASFI rules and implement their own rules and norms, including codes of conduct and finer-grained policy.

Communities indeed exercise their autonomy to implement policy above and beyond ASFI policies. For example, while ASFI encourages projects to admit consistent contributors, the specific process and expected standards for admission are left to each project community itself. [1] For another example, ASFI does not dictate any specific code management protocol and allows projects to institute their own, which conventionally follow one of two approaches: review-then-commit (RTC), in which contributed code is reviewed by a project before it is formally committed to the repository, and commit-then-review (CTR). Consider the following email from Apache Netbeans dated 9/13/2017, excerpting deliberation among Netbeans developers as they discuss their options, clearly informed by the choices of other projects. This illustrates one among many instances of how a range of routine, governed activities emerge in a polycentric system from the coexisting governance structures and may bear varying degrees of influence and independence from the foundation.

> different ASF projects have different policies. The important part is that we should have a common understanding about our commit policy. there might e.g. be a branch for the next release where RTC (review then commit) is applied. That's useful when preparing a release or for maintenance releases we still actively maintain. and beside that we might have a 'future' branch (e.g. on master) or multiple feature branches where CTR (commit then review) is standard. most ASF projects have the whole repo on ctr...

As developers operate within the environment of ASFI's policy, they are expected to increasingly signal their relationship and interdependence with the organization through their behavior, as observable in their discussions and actions [49, 50]. The influence of the foundation's policies on a mentored project's routine governed activities indicates how its governance is being internalized in the community. The more community members discuss policies and incorporate formal elements such as specific offices, requirements, or guidelines into all systematic governed activities, the more we can argue that members have internalized the ASFI formalization. This further motivates us to understand the impact of foundations on projects through policy internalization in conversations about sustained community practices.

[1] https://incubator.apache.org/guides/ppmc.html, accessed 02/18/24

4 RESEARCH QUESTIONS

Situated in the backdrop of OSS-foundation polycentricity, this section presents our research questions, which look at community governance and policy internalization across the different aspects of ASFI governance.

The formalization of governance in traditionally volunteer-driven communities has been a contentious theme. OSS pioneer Eric Raymond observed that the "number of hoops" or too many formalized procedures and rules may drive away skilled contributors [72, 78]. Extensive regulation may introduce additional requirements and necessitate performing more institutional obligations. Therefore, communities may be expected to show more governed activity in domains that are heavily policied, given their presumed importance in the ASFI ecosystem. As a result, we may expect a positive relation between the number of policies and the frequency of observed routine activities in a particular area of governance.

While there are concerns about redundant routines and organizational overheads, lack of regulation may cause individuals/ communities to draw upon larger social and cultural constructs for predictability. Such "tyranny of structurelessness" may perpetuate broader social inequalities [24]. The idea of "green tape" encapsulates the potential of policy to provide clarity and certainty, focus organizational attention, and convey legitimacy [19]. Implications may also extend to OSS formalization, whereby extensive yet well-designed policies may streamline rather than divert developer efforts. Moreover, in governance domains where regulatory clarity is limited, greater project activity may become necessary to sustain development.

This brings us to RQ1, which explores how the focus of formal policy-making in OSS incubators is associated with the observed distribution of governed activities across different governance domains. We identify governance concerns/topics actively shared between the ASFI and its projects through policy documents and extensive mailing lists across 208 communities. Since foundation policies and community governance mutually structure the routine behavior of projects, we aggregate all similar activities concerning these governance topics from email conversations and examine their correlation with the topical distribution of ASFI policies.

RQ1: How does Incubator regulation relate to community-level governed activities across different governance topics?

Organizations engage in many functions, some of which are more critical than others. Institutions manifest through the practice of routines formalized by established rules [44]. Well-designed rules seek to reduce uncertainty and can act as formulaic precedents to replicate success across mentored projects [56], or at least help standardize the provision of ASFI's resources. Hence, more important functions may be marked by a greater amount of policy to formalize behavior and may elicit greater internalization. Consequently, we might expect alignment between the amount of formal policy on a topic and how resulting policy prescriptions are internalized in practice.

On the other hand, activities and related exchanges in a topic may deliberate policy to only an extent, while their actual operations may reflect a marked departure or even be autonomous of formal structure [21, 95]. This may be especially true when certain institutional obligations are ceremonial or necessary to maintain affiliation with the ASFI but are less relevant in day-to-day development. If such is the case, the observable policy internalization among communities across different governance topics may not be correlated to the extent of policy overseeing the topic. Moreover, if policy extent is driven more by the complexity than the criticality of a governance subject, such may paradoxically generate a greater quantity of policy for its various cases and also less internalization, as practitioners take license from that very complexity to exercise greater discretion in how they execute.

RQ2 explores how the extent of policy-making among different governance topics relates to the formal policy internalization among projects. For all topical governed activities, we measure policy internalization in terms of how discourse about those activities in general semantically reflects the policies formalizing those activities. Finally, we examine how such internalization varies with the extent of regulation across topics.

RQ2: How do the levels of policy internalization in governed activities relate to ASFI policy extent across different topics?

For an Incubator program to realize its goals, it is important to assess the association between its governance and project outcomes. At the same time, it becomes equally important for aspiring communities to understand behavior associated with communities that succeed in Incubator programs, particularly the extent of community governance as well the impact of foundation governance on such operations.

ASFI lays down three primary criteria to determine if a project has potential and is capable of sustaining development: 1) there is community activity evidenced by at least two releases, 2) the releases are compliant with the Apache license, and 3) the communities demonstrate sufficient diversity, with committers drawn from at least three entities (companies, research groups, etc.) [23]. The remainder of the policies serve to help the project achieve those goals.

While RQ1 and RQ2 measure *if* there is a relationship between formal policy and community governance, RQ3 uses an externally valid measure of project outcomes to determine whether there *should be* a relationship, i.e., whether communities align governance focus or internalize policies in topics with more formal rules, in order to successfully realize their objectives. In particular, it examines if community governance efforts or the adoption of policies around formalization correlates to their graduation odds in the ASFI.

We pursue RQ3 through a project-level regression of all governed activities (frequency of structured, routine operations) among individual projects alongside the policy internalization among such operations (semantic similarity of governed activities to policies) against a binary measure of project success (graduation/retirement from the Incubator).

RQ3: How do governed activities and the extent of policy internalization relate to the success of projects?

5 DATA AND METHODS

5.1 Variables of Interest

5.1.1 Governance Measures. Philosophers of language such as Austin and Searle have long recognized that speech is action [2, 81]. This is particularly the case in online communities, where so much action is remote interaction. We pursue two discursive measures of community governance from developer conversations in mailing lists, namely governed activity and internalization of policy. Traditionally public and open access, OSS mailing lists are key to collaboration as they promote transparent peer review [46] and solicit reciprocal contributions [67]. Unlike issue tracking and version control logs, these also contain exchanges beyond technical development, such as product planning, community management, ratification of major decisions, licensing, etc. Further, due to explicit ASF policies, all project activity is comprehensively archived across public mailing lists ("If it didn't happen on the mailing list, it didn't happen" [106]).

We described in Section. 3 how routines reflect all prevailing governing norms among projects. We first identify the different governance concerns shared between projects and the Incubator by means of topic modeling of policies and conversations and represent the following two measures by project and governance topic:

Governed Activity: The total number of recurring or routine activities [11] about a governance topic, as discussed in a project's mailing list. A higher presence of governed activity indicates greater governance efforts to structure and routinize community operations. For example, if a community establishes a norm for ratifying releases, future releases will likely follow the established schema. In ASFI projects, such governance is a culmination of the foundation's policies as well as the underlying codes and norms of the community developers. Recurring activities are aggregated over their textual similarity.

Policy Internalization: In the context of an institution, policies are prescriptions whose constituents are the specific actor(s), certain activities they are required to perform towards other entities and objects, under a particular context [15, 86]. As mentored projects integrate themselves into the foundation, their operations and discussions are expected to take after ASFI policy by embedding the roles, responsibilities, and activities it defines into their own operations. Policy internalization reflects the extent to which activity in an institution is actually being structured by policy. Taking an assumption that one's exchanges within an institution reflect their mental models of it [49, 50], this measures to what extent and in what manner activities are being mentally represented, and in that sense internalized, in the terms of what formal policy provides for them.

Semantic similarity is an assessment of meaningful and conceptual relationships between texts [43]. For a topical governed activity in a project, we measure policy internalization through its semantic similarity against policies within the respective topic.

Policy Extent: A foundation-level variable indicating the extent of ASFI's regulation across topics. It is represented as the frequency (count) of formal rules overseeing each governance topic, with higher values (number of rules) in a topic indicating greater ASFI regulation.

5.1.2 Project Membership and Activity. Projects in ASFI are diverse, and their governance and Incubator outcome may also be subject to community structure, activity levels, etc. Since we are interested in analyzing how governance behavior correlates to project sustainability, our analysis has to simultaneously control project attributes, such as community size and development intensity. We incorporate four suitable covariates in our analysis through *community size (committers), number of commits, code base size (lines of code; LOC)*, and finally, the frequency of interaction among the project developers (*developer emails*) over mailing lists.

5.2 Datasets

We center our analysis of ASFI governance through a set of 234 comprehensive policies, which were coded across the key ASFI documents and guidelines [82]. These span multiple sources such as the official Apache Incubator Policy manual, the Community Guide, the Podling Project Management Committee (PPMC) Guide, the Apache Cookbook, the Mentorship Guide, the Graduation and Retirement Guides, and finally, the Release Management Guide.

In the ASFI, project incubation lasts up to several months, followed by an assessment and a formal vote to decide on its graduation into ASF for continued support or retirement. Yin et al. scraped all mailing lists across 269 Apache projects from when they joined the Incubator and up to their last day in the ASFI [106]. Since we solely focus on norms and activities within communities, we only retain the 'dev' (community developers) subdirectory emails across all projects. We exclude redundant content such as auto-generated emails for issues posted and resolved and other development-related notifications (JIRA, Github) through source address-based filtering. Periodic emails were also circulated by the Incubator Project Management Committees (IPMC) or project mentors, which were formal, administrative, and generally concerned with progress reporting. All such emails have a fixed format and were identified and filtered through string matching. This mitigated potential bias in measurements due to superfluous policy content from the administration, as our subsequent analysis concerns governance-related behavior within and among community developers only.

For project-level covariates, we obtain commits, lines of code, and the number of active contributors. ASFI projects use GitHub, Subversion, or a combination of both to maintain their codebase. Stănciulescu et al. [89] extracted monthly performance metrics for 218 ASFI projects through their incubation. However, the tooling infrastructure they developed only supported mining software metrics from Git repositories. Moreover, Yin et al. mined project mailing lists up to Jan 2021, including ones that were mostly SVN-based, while Stănciulescu et al. span projects from March 2003 up to May 2021. Given these differences, we based our study only on those projects that are common to both datasets. This yielded 214 projects for which both project measures and email data were available.

Moreover, there were some differences in the way these data were collected. Yin collected data in time windows of 30 days, whereas the other dataset collected data on a monthly basis (calendar timestamps). To resolve this mismatch, we modified the collection timeline to a 30 days time window in the tool provided by Stănciulescu et al. to match the time window in the dataset from Yin et al. and repeated the measurements for our variables of interest for these 214 projects.

5.3 Measurements

5.3.1 Extracting activities. Routines have been studied at multiple levels, from the most nuclear activities to complete processes. The

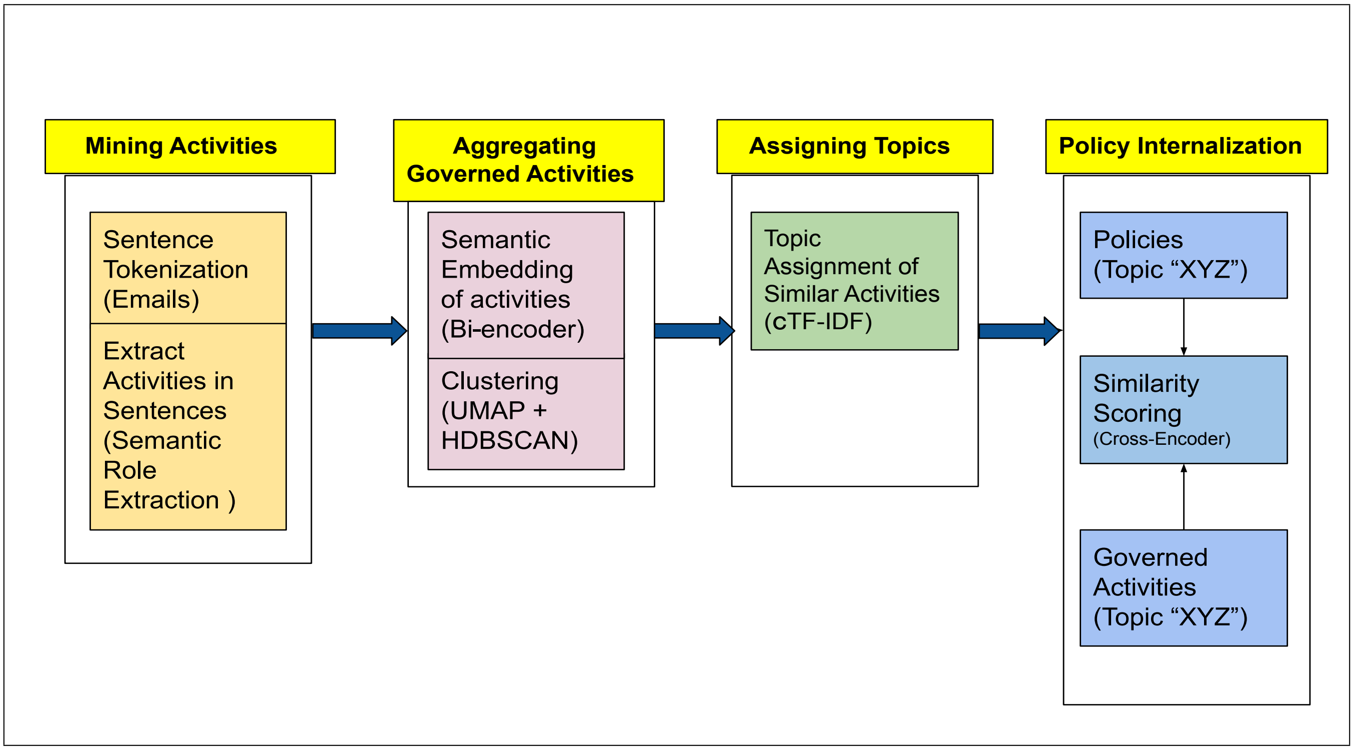

Figure 1: Language modeling pipeline for extracting activities, aggregating routine governed behavior, and evaluating internalization.

most fundamental unit, the performance program [51, 69], is defined as a 'chunk' of scripted activity, generally a routine in itself or part of a larger process. To capture organizational routines from ASFI email discourse, email texts and policies were first tokenized into sentences through StanfordNLP's Stanza library [70]. We next turn our attention to extracting different activities from within these sentences.

This serves several purposes. Firstly, most existing language models, including ones subsequently used, encounter complexity overheads and truncate long sentence inputs beyond a certain token sequence length. Secondly, sentences can be compound, conveying multiple activities with their specific context and possibly spanning different topics (Table. 2). Therefore, decomposing sentences into granular units of analysis, like performance programs, allows depth and insight in subsequent analysis.

We decompose sentences while preserving their context. Context is important in understanding different routines and their place in the development ecosystem (E.g. *'Projects **issuing** press releases'* vs. *'Resolve **issues** that are release blockers'* or *'Projects **requesting** Apache infrastructure'* vs. *'Project Management Committee **requesting** progress report'*). To attain fine-grained extraction of different activities and their context nested within sentences, we use semantic role labeling.

Semantic role labeling or SRL [30, 43] is an NLP task that extracts roles (actors, direct or indirect objects, etc.) associated with an action (verb) along with other modifiers from a sentence. Additionally, SRL also extracts constituents with contextual information such as the time of act, manner, direction, goal, purpose, cause, etc. [4].

Original Policy:
'After a vote has finished, the ipmc must send a notice email to the board and then wait for 72 hours before inviting the proposed member'

Semantic Role Parsing:
'ARG0': ['the ipmc'], 'ARGM-MOD': ['must'], 'V': ['send'], 'ARG1': ['a notice'], 'ARGM-DIR': ['email'], 'ARG2': ['to the board'], 'ARGM-TMP': ['after a vote has finished']
'ARG1': ['the ipmc'], 'ARGM-MOD': ['must'], 'V': ['wait'], 'ARGM-TMP': ['after a vote has finished', 'then', 'for 72 hours', 'before inviting the proposed member']

Performance Programs (After reconstitution):
'After a vote has finished the ipmc must send a notice email to the board'
'After a vote has finished the ipmc must then wait for 72 hours before inviting the proposed member'

Table 1: Activities extracted from compound sentences through Semantic Role Labeling (SRL). ARG0 denotes agent, ARG1-ARG6 are direct/indirect objects, ARGM-MOD indicates modals, while ARGM-TMP and ARGM-DIR are the temporal and directional arguments, respectively

We chose a BERT-based [20] implementation of SRL [84] developed by AllenNLP [27] on the Propbank annotation scheme. The model holds a state-of-the-art performance on the English Propbank

Original Sentence: '(1) I'll be away from my computer starting Friday and through the New Year, so I won't be able to do much to help if folks want to release 2.1 during that time (not even testing).' (Apache Roller, 12/21/2005) **After SRL and reconstitution:** 'I'll be away from my computer starting Friday and through the New Year' (Schedules/Events) I won't be able to do much to help if folks want to release 2.1 during that time (not even testing)' (Release Management)

Table 2: Capturing granularity: Sentences spanning multiple, thematically distinct operations. In this example, a developer shares their vacation timeline with the community in general while also discussing implications for a tentative release. Topics indicated for each activity are inferred as described in Section. 5.3.3

(Newswire) as well as a test F1 score of 0.864 on the Ontonotes 5.0 dataset. We identify all possible semantic roles associated with each distinct verb from compound sentences. These SRL frames were reconstituted into distinct activities by reordering the semantic roles and all other contextual arguments for each verb along their relative positions from the original sentence. The 723,863 developer emails across the 214 projects generated 2,248,950 expressions of activities after email filtering (Section. 5.2), sentence tokenization, and SRL-based parsing.

In governance research, rules are specified in terms of grammatical constituents representing the governing (committees, boards, etc.), the governed (e.g., committers), the activities they undertake, and the conditions they entail (e.g., voting before a release) [15]. Our policy reference data [82] comprised descriptive policies spanning multiple nested rules (Table. 1). Therefore, SRL-based preprocessing was also extended to the policy documents, whereby the 234 policy descriptions from Sen et al. were parsed into 422 individual rules.

Finally, we conduct an additional pre-processing step. Developers often use mailing lists for technical discussions and clarifications. As a result, they often contain stack traces, logs, etc., which may be parsed as regular activities. We restrict our analysis to human-readable, standard English-language data, which can be compared and interpreted against governance policies such as those of ASFI. We detect and retain only English texts using a HuggingFace XLM-Roberta-base model [13] trained for language identification. This reduced the number of extracted activities to 2,029,691.

5.3.2 Governed Activities: Aggregating routines. As described in Section. 3, routines are activities carried out time and again under specific circumstances [11]. Unlike well-documented formal policies, routines are more dynamic and span activities dictated by emerging norms and operational priorities. Hence, it is extremely challenging to comprehensively codify activities in a community and train models that can discriminate routine behavior from non-routine ones.

Importantly, we are interested in a pipeline that supports governance analysis across diverse online communities. Since routines are influenced by technological trends, the nature of the product, the specific community, utilities involved, etc., there may arise inaccuracies in inference when extending a supervised model specifically built on ASFI data to study behavior among other communities and foundations [85]. Therefore, based on theoretical definitions of our construct of interest (i.e., governed activities are routine or 'recurring' operations), we leverage alternative learning methods compatible with our goals. We hereby describe our approach to discovering routines as similar activities in email data through semi-supervised clustering.

We find similar ('recurring') activities through semantic similarity-based aggregation [74]. Popular approaches to semantic representations include word-level [58, 68], sentence-level [10, 14, 98], and more recently language model-based approaches which allow for more advanced representation learning for different semantic tasks. We use a general-purpose bi-encoder [74] pre-trained on a domain-relevant corpus from Stack Overflow [12], a question-answer platform specially used by software developers. We used the model to generate semantic embeddings for all the governed activities we extracted.

Next, for aggregating the encoded texts, we use BERTopic [29]. It supports hierarchical density-based clustering or HDBSCAN [55] for most bi-encoder model embeddings, followed by topic modeling of the inferred clusters. To train the clustering model, we uniformly sample 100,000 activities out of all the 2,029,691 activities previously extracted. Modeling activities across projects together allows for identifying and grouping them under a set of shared governance topics.

To cluster community activities intersecting with ASFI concerns, the 422 rules from ASFI policies are passed as seeds to BERTopic to initialize the embedding space. For best clustering results, we conducted hyperparameter tuning for BERTopic's HDBSCAN over the density-based clustering validity (DBCV) metric [60]. We conduct a thorough grid search over several hyperparameters for the optimal training configuration and retain the model with the best relative DBCV score (0.32 on a scale of -1 to +1). Details of the process are provided in the Appendix. B.

5.3.3 Topic modeling of governed activities. BERTopic finally conducts cTF-IDF, an adaptation of TF-IDF [77], across the dense clusters of governed activities to select their most representative words. We use these words to identify appropriate themes among the clusters. Topic coherence metrics [75] supported by Gensim [73] evaluate topic modeling performance on a scale of 0 to 1. Our final model shows a topic coherence C_v of 0.683, indicating strong coherence between the content of the topics and the representative words. Details on the C_v metric and its interpretation are included in the Appendix. B.

Of the 422 rules parsed out of the 234 policies, 106 of these were not found to be related to any significant cluster of activities and were disregarded as topical outliers by BERTopic. Around a third of all activities were also excluded as non-routine outliers. A total of 211 topic clusters were discovered among the non-outlier activities, among which 42 were identified as common between the 316 ASFI rules and email activities. Around 493,008 activities were found to belong under these 42 governance topics, i.e., topics associated with at least one ASFI rule. Final topic label assignments were deduced based on the assigned policies, activities, and top keywords from each topic, and overall domain knowledge of the ASFI. Detailed descriptions for these topics are provided in the Appendix. C.

5.3.4 Measuring institutional internalization. Policies address an important subset of organizational possibilities, and any institution will have a large degree of overlap between the activities governed by policy and those that are performed by participants. However, assessing qualities of this overlap, such as divergences between policy and practice, is traditionally difficult. For governed activities under any ASFI governance topic, we measure the extent to which they reflect policies overseeing the respective topic.

Prior work has studied enculturation and employee exit by treating individuals' linguistic divergences as a measure of cultural fit in organizations [87]. This depends on basic assumptions of cognitive science that the words people use to describe something are a reflection of their mental representation of such. Hence, a change in word usage reflects a change in representation, and entities with similar usage have similar mental representations [34, 96]. While Srivastava et al. compare the speech of high- and low-power members of the organization, we take on the more difficult task of crossing modalities and comparing the speech of members about work activities to the policy texts governing them. This required improvements in methodology, from computing lexicon-based similarity to conceptual similarity through semantics.

Measured on a continuous scale, semantic similarity is a comprehensive measure that accounts for not just words present in texts but the overall meaning they convey together. Developers, besides discussing policies in form, also act and discuss policy aspects situated in their practical circumstances. Therefore, semantic similarity can be employed to quantify levels of internalization around the constituents of policies among observed activities based on how they invoke roles, designate responsibilities, and introduce requirements by formalization. This is important in order to represent graded changes in behavior along the rates of institutional diffusion [88]. For example, observations that a group of developers discussing software releases will, over time, increasingly use words and constructs as they are employed in ASFI software release policies are important to record how institutions are gradually internalized. The lowest values of internalization are assigned to activities that have little in common with the constituents or overarching context of a policy. Moreover, the semantic similarity task is designed in a way to be able to detect negation within texts with similar contexts [9, 99]. Consequently, it is able to interpret activities that are situated in the same formal contexts and dwelling on the same policy concerns as compliant activities but are contradictory to policy and assign them relatively lower values of internalization.

Cross-encoders and poly-encoders are standard language model architectures that treat sentences or text to be compared as simultaneous inputs and apply attention [94] over them jointly for semantic comparison [36, 74]. Bi-encoders and cross-encoders are often used together for information retrieval and text ranking. While bi-encoders can encode individual sentences to support high-level clustering over large sets of text, cross-encoders are suitable for more precise, pairwise comparison between smaller sets of texts [91].

We use a DistilRoBERTa-base cross-encoder [92] from Huggingface, which rates text pairs on a continuous scale of 0 to 1, with higher scores indicating greater similarity. The model demonstrated a Spearman rank correlation of 0.87 with respect to the human-annotated scores from the STS text similarity benchmark [9]. Using this cross-encoder, we compare every governed activity against all the rules assigned to the same governance topic to find the ones it resembles most closely through the maximum pairwise semantic match. The mutual semantic similarity score between a governed activity and the closest policy is used to represent the speaker's *internalization* of ASFI formalism through the activity's relatedness to policy. Through this method, we obtain internalization scores for all the 493,008 governed activities under each of the 42 governance topics.

Table. 3 provides illustrative examples of how we evaluate the practical observance of policies in developer behavior along the dimensions of entities, objects, activities, and associated conditions described in formal policies. We look at governed activities and their closest policy match from some of the most highly regulated (PPMC, Emails, and ASF/Contributor License) and high activity (Releases, Issues) topics. Internalization is high for activities that communicate the same concept as a policy. It becomes lower depending on the manner in which developers reinterpret or act with respect to the stated policy. E.g., in 'Email Communication", Apache Esme enforces mailing lists for all-inclusive deliberations in the Incubator beyond simply development. In the topic 'Release Management', where policies require legal conformity for an Apache release, we see a developer clarifying whether there are additional technical standards as well. In 'Apache Foundation/ Contributor License', we see activities that are compliant with those described in ASFI policies on release ratification. One of the activities in 'Issues' depicts how developers incrementally work with disclaimers and associated artifacts toward legal compliance over incubation. As expected, we also observe low internalization for activities that are contrary to or critical of a policy, for e.g., prioritizing graduation over release, unvoted artifacts or even questioning the decisions of the PPMC governing body.

Internalization decreases significantly as activities have increasingly less in common with the specifications of a policy. E.g some communities use a particular mailing list management norm to redirect between lists ('Email Communications'), a practice that ASF policies on email communications do not speak to. Activities around describing issues, troubleshooting, and discussing solutions span institutional and technical concerns. Several of these deal with niche technical instances not addressed by policy and, therefore, have low internalization. The activity in 'Issues' with the lowest internalization score describes a persisting issue with a specific component, thus differing from the policy in terms of both the issue-related activity (successful issue resolution) and the type of issue (development than licensing) it depicts.

We converged upon semantic similarity following an extensive review of NLP methods that can realize our construct operationalization. Natural language entailment [6, 100] was the closest other semantic subtask that could be potentially used to determine whether an observed activity was in agreement/disagreement with policies. Such binary operationalizations were, however, found to be insufficient to account for the drift between formally articulated statements (framed policies) and informal, practical discourse (conversations) or grade rates of formalization in activities and were thus inadequate for representing internalization as we define it.

Topic: Release Management **Policy:** "Producing fully-conformant releases is a condition for graduation"	
Governed Activity:	Internalization:
A release is a pre-requisite to graduation as that is the point at which the ASF is able to assert that the code is fully license compliant. (OpenOffice, 2012-03-13)	0.75
Is having a completely automated release process a hard requirement for graduation? (Usergrid, 2015-01-22)	0.61
We need to graduate * before * we are ready to release 1.3. (Wicket, 2007-04-26)	0.47
Topic: Issues **Policy:** "By the time you graduate all issues listed in the disclaimer must be corrected"	
Governed Activity:	Internalization:
I just fixed a couple of very minor issues (a file missing ASF header, missing disclaimer and a couple of copyrights) (Rocketmq, 2016-12-20)	0.60
Looks like all of the issues mentioned in the description for this bug have been handled now, so I give this a +1 to be committed (Derby, 2005-06-21)	0.37
All other commands work but I'm trying to get VM console which just not work (Cloudstack, 2013-02-08)	0.04
Topic: Email Communications **Policy:** "Development should happen on the official mailing lists at Apache"	
Governed Activity:	Internalization:
Development within Apache must take place in the open, on the mailing lists, through a central repository. (Buildr, 2008-05-01)	0.83
I'd just like to remind people that in Apache projects all decisions and discussions must happen on the mailing lists. (Esme, 2009-01-05)	0.60
Can you please pin it to the top which directs people to the apache mailing lists? (Joshua, 2016-04-11)	0.44
Topic: Apache Foundation/Contributor License **Policy:** "When a project decides it wants to make an ASF release the project must hold a vote on their public dev list"	
Governed Activity:	Internalization:
Any ASF release needs to be voted upon. (Roller, 2005-09-02)	0.82
Once the vote passes according to the ASF's voting rules, the release manager makes the necessary announcements usually after waiting for the artifacts to be propagated to all mirrors (Wicket, 2007-02-10)	0.56
But the unvoted source tars make me unsure whether this is acceptable in the ASF (Echarts, 2020-08-03)	0.43
Topic: Podling Project Management Committee (PPMC) **Policy:** "The PPMC is directly responsible for the oversight of the project"	
Governed Activity:	Internalization:
The PPMC is directly responsible for the oversight of the project (Zeppelin, 2015-11-28)	0.99
One of the PPMC's primary responsibilities is to make sure that the project remains in compliance with the legal & foundation policies (Streams, 2012-12-18)	0.59
How does the PPMC provide proper oversight in considering someone's merit without work here at Apache? (OpenOffice, 2012-03-02)	0.45

Table 3: Governed activities across topics and their Policy Internalization

5.4 Analysis

RQ1 and RQ2 pursue an ASFI-level exploratory analysis of our governance measures along the policy extent. RQ1 compares the proportions of ASFI rules (level of regulation) and project-level governed activity across the topics, while RQ2 follows up by assessing the distribution of ASFI policy internalization in activities. Finally, for RQ3, we examine governance behavior among projects against their graduation or retirement from incubation. We set the significance level of our analysis at the standard $p < 0.05$.

While we evaluated governance measures for the 214 projects common between the email [106] and the project performance datasets [89], six projects were further dropped prior to all analysis as the metrics tool [89] could not reliably measure all their commit history. Therefore, we perform our analysis and report our findings on 475,546 governed activities from 208 projects across the 42 ASFI governance topics.

5.4.1 RQ1: How does Incubator regulation relate to community-level governed activities across different governance topics? As described in Section. 4, we focus our analysis on governance topics shared between the ASFI and its mentored projects. We provide a comparative visualization (Figure. 2) and evaluate the Pearson correlation between the ASFI's policy extent through the 316 rules assigned to 42 governance topics against the distribution of governed activity along the same topics.

5.4.2 RQ2: How do the levels of policy internalization in governed activities relate to ASFI policy extent across different topics? Higher mean internalization scores indicate that in a particular topic, the

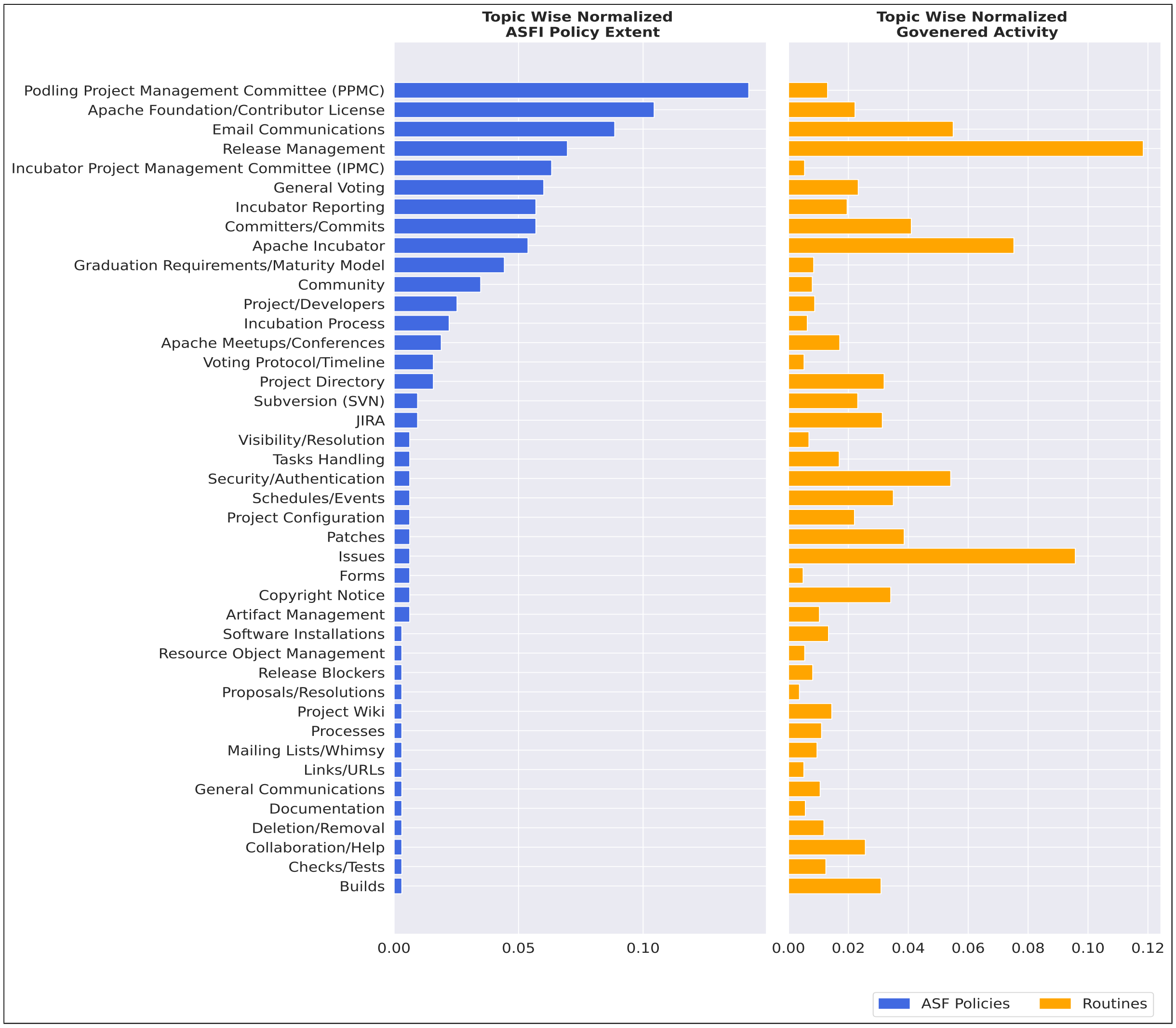

Figure 2: Left: ASFI policy extent as the normalized distribution of 316 rules over the 42 governance topics they represent. Right: Normalized distribution of 475,546 governed activities from 208 projects across the same topics. Governed activity was not found to be significantly correlated to policy extent.

projects' practiced routines are more framed by formalized Incubator policy. To explore RQ2, we additionally examine the distribution of internalization scores of governed activities conditioned on governance topics (Figure. 3) and evaluate the Pearson correlation between the topic-wise policy extent and mean internalization scores across topics.

5.4.3 RQ3: How do governed activities and extent of policy internalization relate to the success of projects? We fit a binomial generalized linear model (GLM) of project-level measurements of governance as well as the covariates against their respective incubation outcome. We conduct our analysis through the GLM suite (regression, multicollinearity check, and validation of assumptions) supported by the *statsmodel* package in Python. LASSO-based variable selection is conducted prior to regression and inference, for which we use the *group-lasso* Python package.

ASFI strives to build meritocratic communities and assesses its projects' performance throughout the incubation time frame. As membership and activity levels undergo constant changes in OSS, we average the monthly measures of active committers, developer emails, and commit activity to capture their sustained levels. The code base variable was represented as the net size of the project repository in terms of overall lines of code (LOC) written by the project while in ASFI. Prior work on ASFI has shown that successful projects tend to graduate early [104], so we incorporate the total number of months spent by the project in the Incubator as one of the covariates. Similarly, to adapt the governance measures we represent governed activity through the total number of routine activities observed in a project during incubation across the

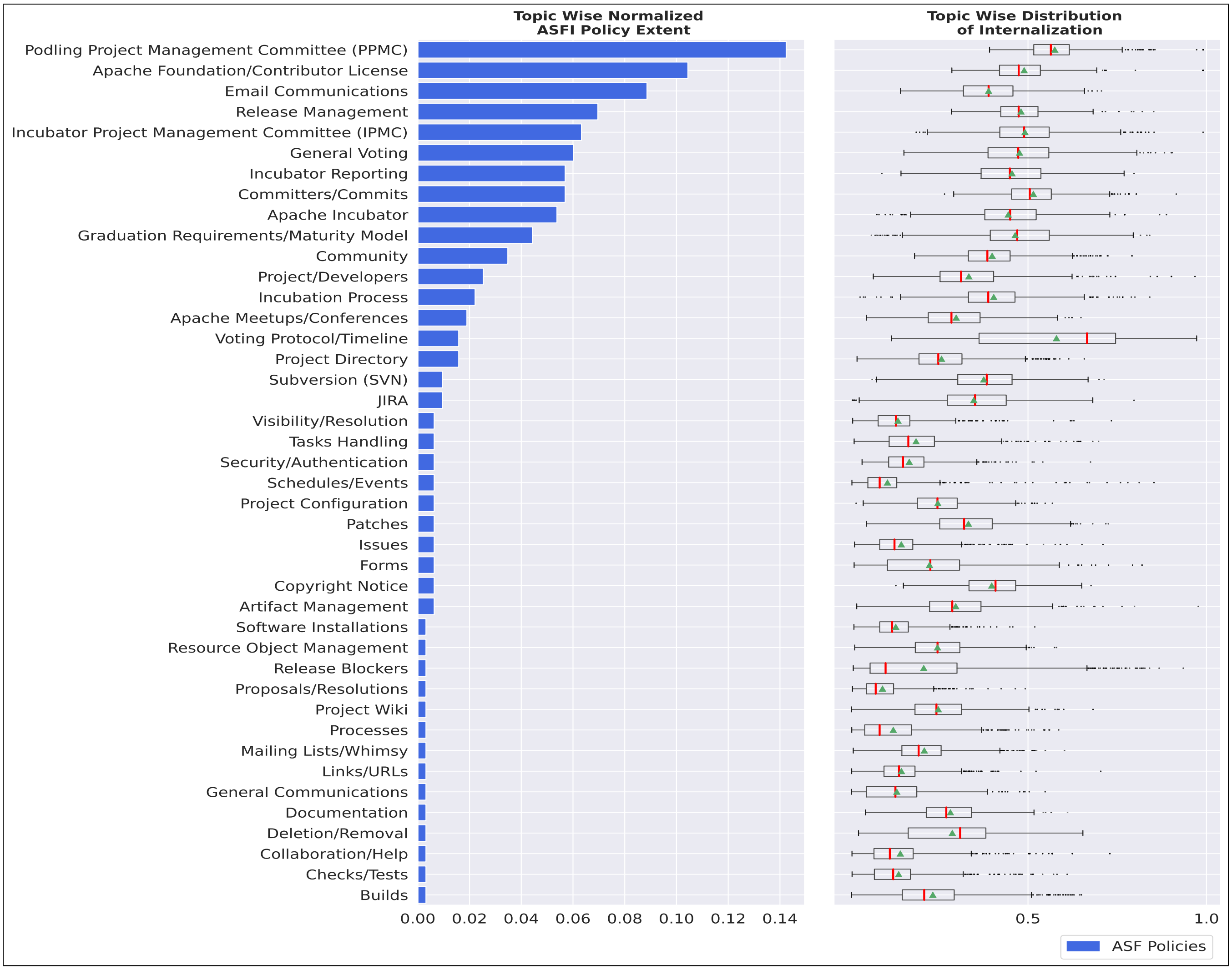

Figure 3: Left: ASFI policy extent as the normalized distribution of 316 rules over the 42 governance topics they represent. Right: Distribution of internalization scores of activities within their topics. Red and Green markers indicate the median and mean, respectively. Internalization is observed to be higher in governance topics which are more regulated.

mailing list. The overall policy internalization along a governance topic in every project was similarly evaluated by averaging the scores across all the governed activities. The resulting number of predictors was 89, including five covariates and the two distinct governance measures from each of the 42 topics.

Certain project mailing lists did not reflect governed activity under some of the topics, making the *governed activity* of that topic equal to 0. There are 54 projects with 0 observed *governed activity* in at least one topic. Rather than dropping those observations entirely, we retained them in a way that minimizes information added to the system through the imputation procedure but allows us to retain the information in the non-missing variables: unmeasured internalization scores were filled through iterative round-robin imputing supported by the Python package Sklearn. This method of imputation, a pythonic implementation of MICE [93], is unbiased relative to other choices we could have made, such as assigning 0.[1]

Project-level covariates (committers, emails, codebase, and commit activity), as well as governed activity for every topic, were log-scaled to address skew and to facilitate comparison along the scale of different projects. Subsequently, all variables were standardized through z-score standardization. We then addressed multicollinearity by removing all variables with Variance Inflation Factors > 5. We then performed a logistic LASSO-based variable selection over 5-fold cross-validation and hyperparameter tuning over

[1]The model was run with list-wise deletion for all projects with at least 1 missing value, and the only difference was a change of the sign for the number of commits, an effect for which we have low confidence in all our models. We further repeated the analysis without 5 topics with more than 10% missing entries. We did not observe major changes in effects (size, direction) and significance.

the log loss. After multicollinearity tests and variable selection, we have a reduced set of 11 significant predictors.

We construct nested linear regressions, whereby we fit four models to assess the contribution from different groups of variables (Table. 4). These are the "baseline" models with only covariates as predictors (M1), a second model adding topical governed activity variables to the baseline (M2), a third model adding only policy internalization variables to the baseline (M3), and the final full model including all three groups of variables: baseline covariates, governance activity, and policy internalization measures (M4). For every model, we additionally checked for outlier influence using Cook's distance and found no data points with extreme leverage ($D > 1$). The assumptions of log odds linearity were validated using the Box-Tidwell test, whereby no interaction terms $x * log(x)$ were found to be significant. We observe that the predictive efficiency and fit of the models improve with step-wise addition of governance variables, a reassuring sign of valid model construction across the three types of variables. The full model M4 was found to be the most parsimonious (ΔAIC = 23.05 with second-best model) with goodness of fit at 0.648 (Tjur's psuedo-R^2). Further, it showed a weighted F1 score and accuracy of 93.6% and 93.7%, respectively. We hereby report our findings based on M4.

6 FINDINGS

Results from RQ1 (Figure. 2) show that overall, policy extent has no significant correlation with the frequency of governed activities observed across topics. The Pearson correlation between the topical distributions of policy extent and governed activity was found to be 0.23 ($p = 0.13$), indicating that how communities perform governed activities across topics is uncorrelated with the amount of policy structuring those topics. Hence, substantial differences prevail between the formal policy-making attention of the ASFI and community governance actually enacted by projects. Yet through RQ2 (Figure. 3), we also find that topics with higher policy extent see greater policy internalization with the Pearson correlation between the topic-wise policy extent and mean internalization scores being 0.744 ($p < 0.001$). This indicates generally greater internalization with increasing policy extent. In other words, areas of governance that receive more attention in formal policy also tend to be enacted by participants in a way closely related to the policy descriptions. Therefore, while project governance efforts and ensuing governed activities do not mirror the distribution of policy across governance topics, the internalization of policies is highly correlated with how much formal policy governs that topic.

In RQ3, we test our governance constructs against project outcomes (Table. 4). Factors that correlate positively with a project's chance of graduating include greater internalization of policies related to "Project configuration", "Graduation requirements/Maturity Model", and "Voting protocol/Timeline." Moreover, projects that govern patch-handling activities, i.e., more governed activity in "Patches", are associated with higher graduation odds. On the other hand, factors that correlate negatively with successful graduation include high internalization of "Project Wiki" and a higher volume of governed activity on Incubator reporting.

We observe that neither governed activity around nor internalization of the five most highly regulated topics (those on committees, licensing, email communications, and releases) predicts project success. In fact, project success seems to be correlated mostly with the internalization of policies that receive little attention in formal policy. This further complements our overall finding that projects do not run how they say they run and suggests that formal policies may not present the full picture of how communities govern to sustain themselves.

Our primary analysis is correlational and not causal. This is important to emphasize because our findings for the "Graduation Requirements" topics are probably a spurious but encouraging validity check: it is likely that the act of a project graduating and conducting necessary protocols explains the positive effect of internalization of graduation policies. Similarly, "Project Wiki" is composed of a policy that is only activated once the Incubator has voted to retire a project. The most likely explanation for its negative effect is that project retirement is causing policy enactment, not the other way around.

To check model robustness and probe some unidirectional interpretations, we perform a post-hoc analysis where we repeat all experiments with a modified policy dataset that excludes these confounding end-of-incubation-related policies that happen after a determination of graduation or retirement has been made. We focus this robustness analysis exclusively on policies that are relevant to the active incubation and growth phase of ASFI projects. Therefore, we removed 34 out of the 234 policies that are generally applicable for projects post-graduation/retirement or only at the terminal stage of incubation (graduation vote, transferring trademarks, or ceremonial protocols of graduation/retirement, etc.). For RQ1 and RQ2, we once again retain the previously observed trend, or lack thereof, between policy extent, governed activity, and internalization. For RQ3, we retain significant effects from three out of the six variables that stood out in our original analysis. These include "Patches" (governed activity), "Incubator Reporting" (governed activity), and "Voting Protocols/Timeline" (internalization). As expected, we no longer observe the significant effect associated with 'Graduation requirements', which comprised several policies (now removed) closely related to the graduation event, while 'Project Wiki', which treated post-retirement project wrap-up, was not among the topics inferred from the reduced set of policies. Lastly, internalization from the topic 'Project Configuration', which contains policies on using ASFI infrastructure, does not exert a significant influence on project outcomes. We elaborate and interpret our findings through observations supported by both the primary and supplementary analyses. Details about the supplementary analysis are provided in the Appendix. A. Table. 5 provides examples of governed activities and policies for these three topics.

Domain knowledge of the ASF Incubator can help us further contextualize the results from RQ3 (Table. 4). Democratic communities and consensus building are encoded in ASF's functioning ('The Apache Way') and are a hallmark of the OSS movement generally. ASFI requires project-level voting for approving releases, appointing members to the project PMC, admitting committers, etc. Observance of ASF's standard voting procedures likely indicates acceptance and shared understanding of established protocol for soliciting consensus in decision-making. Projects that have high internalization of policies regarding "Voting protocol/Timeline" are

	Covariates Only		Covariates & Governed Activity		Covariates & Internalization		All Variables	
Predictor	Coefficient	p	Coefficient	p	Coefficient	p	Coefficient	p
Intercept	2.490	0.000	3.032	0.000	3.252	0.000	4.427	0.000
Committers[2]	0.077	0.874	-0.018	0.973	-0.3074	0.637	0.127	0.875
Commits[2]	0.705	0.140	0.615	0.243	0.772	0.195	0.197	0.793
Developer Emails[2]	0.807	**0.016**	1.069	**0.020**	1.000	**0.020**	1.188	0.079
Incubation time[1]	-0.518	**0.011**	-0.181	0.555	-0.799	**0.004**	-0.334	0.420
Incubator Reporting[2]			-1.210	**0.011**			-1.827	**0.002**
Patches[2]			0.688	**0.011**			1.009	**0.009**
Project Configuration[1]					0.765	**0.002**	0.623	**0.043**
Task Handling[1]					-0.511	0.054	-0.609	0.084
Project Wiki[1]					-0.720	**0.032**	-1.417	**0.005**
Voting Protocol/ Timeline[1]					0.428	0.129	0.933	**0.013**
Graduation Requirements/ Maturity Model[1]					0.898	**0.001**	1.058	**0.002**
Observations: 208	R^2 (Tjur): 0.258 AIC: 139.91		R^2 (Tjur): 0.360 AIC: 124.96		R^2 (Tjur): 0.486 AIC: 113.34		R^2 (Tjur): 0.648 AIC: 90.29	

[1] Standardized variables [2] Log transformed (base 10) and standardized variable

Table 4: Summary: Binomial (Logit) GLM regression of project governance against Graduation/Retirement

Topic: Voting Timeline **Policy:** "A majority vote lasts at least 72 hours"	
Governed Activity:	Internalization:
This vote is open for at least 72 hours (Falcon, 2014-11-05)	0.80
Provided there are no -1s this vote will be open for 72(+) hours (Abdera, 2008-03-25)	0.60
The vote will run for at least 120 hours. (Hawq, 2017-04-01)	0.39
Topic: Incubator Reporting **Policy:** "Projects shall report monthly for their first three months after that quarterly"	
Governed Activity:	Internalization:
New projects need to report monthly for the first three months then quarterly after that. (Impala, 2016-02-10)	0.89
We already submitted 3 reports. (Lucy, 2010-11-01)	0.51
We missed a report for one month (CouchDB, 2008-04-03)	0.27
Topic: Patches **Policy:** "Promptly reviewing patches or pull requests is essential"	
Governed Activity:	Internalization:
Reviewing the patches in a timely manner is key. (Wookie, 2009-11-12)	0.77
Just wanted to check how you are progressing on the patch update, following comments by myself and [NAME] (Derby, 2005-03-11)	0.48
Printing all updates on the website causing the contributor patches to be watered-down/less visible. (Jspwiki, 2013-03-31)	0.15

Table 5: Examples of governed activities from the main effects in RQ3. 'Voting timeline' contains the important ASF-wide standard practice of allowing a minimum of 72 hours for any majority vote. Compliant voting events result in high internalization. Voting may sometimes be suspended before the announced 72 hours in case of a single veto. Reasons for votes longer than the usual 72 hours include community activity levels or major holidays. 'Incubation Reporting' includes discussions around preparing status reports for the ASFI board, while 'Patches' captures a range of operations around reviewing and managing patches.

successfully hosting and running those votes according to ASFI requirements and mobilizing community participation along standard and accepted timeframes.

We find a large negative relationship between the frequency of activities around "Incubator reporting" and the likelihood of graduation. We further investigate and find that projects generally discuss and work on reports only when they are due, except when they 1. miss a deadline and are assigned a new report date, 2. need to keep working to resolve issues in a submitted report, 3. are struggling and asked to report more often.[2]

[2]https://incubator.apache.org/guides/ppmc.html#podling_status_reports, accessed 02/17/2024

Projects often lag in reporting when their development stalls and the community is struggling. In such a situation, the ASFI intervenes actively and necessitates more efforts to motivate the projects to meet standards and resume compliance with Incubator requirements. Therefore, the effect is likely associated with struggling projects and how the Incubator interacts with them. If this interpretation holds, the mechanism for our correlative findings is that an outside factor ("struggling project") is driving both the independent variable (more reporting) and the dependent (reduced graduation chances).

7 DISCUSSION

Our goal was to investigate the relationship between formal policies overseeing OSS communities and their actual self-organizing tendencies. OSS-supporting foundations create policies to encode their concerns and priorities. ASFI introduces formal hierarchies through various offices and committees to organize traditionally free-form OSS communities. They also include requirements to ensure standards of development and conduct among projects.

Governed activities or routine operations indicate the extent of community governance. Structured activities along a governance topic indicate how developers coordinate and conduct the bulk of their activities from their underlying beliefs and current needs. Therefore, more governed activities are expected as a community seeks to structure and routinize more of its operations.

As communities undergo formalization, their governance may be expected to reflect their overarching policy focus. The conventional perception of OSS formalization anticipates more institutional formalities and obligations (Section. 4). This may be observed as increasing community attention on domains on which ASFI sets more rules and routine activity from such structuring. RQ1 tests whether the attention of community governance, as observed through their routine activities, aligns with that of formal policies across shared governance domains.

While governed activities reflect the extent of community governance across topics, we are also interested in how communities align formal rules and actual governance behaviors. In their efforts to structure activities, projects may choose formal policies, implement their own norms, or a combination of both (Section. 3). RQ2 further examines if the extent of formal regulation is related to how community governance integrates them, as observable through the policy internalization of governed activities.

Our results from RQ1 (Figure. 2) indicate that the extent of ASF's regulation does not, in general, seem to increase the intensity of "on-the-ground" governed operations proportionally. At the same time, our findings from RQ2 (Figure. 3) suggest that through extensive policy-making along specific concerns, the ASFI succeeds in using policy to orient community governance, which shows up through policy internalization in governed activity along domains with more extensively defined policies.

We reconcile the implications of the two approaches to understanding formalization. RQ1 dwells on convergence and divergence in the ASFI's community governance efforts, i.e., formulating, establishing, and implementing rules and norms to structure activities. Meanwhile, RQ2 examines the extent to which community governance incorporates the ASFI's formal policies: literally how much communities internalize a formal policy's framing of a governance issue. The positive correlation between internalization and policy extent likely indicates that certain governance topics that are extensively codified considerably structure governed activity. Yet results from RQ1 indicate that highly formalized governance topics elicit relatively less or no more governance effort from communities as compared to those where fewer formal rules exist. In fact, in several crucial topics with limited regulation, projects exert substantial governance effort to sustain operations. The takeaway is that the effect of more formalization in policy seems to be reflected less in the volume of governance activity it spurs and more in how closely that activity hews to prescribed standards.

The ASFI's policy coverage is largely administrative and outlines appropriate protocols for governance concerns it deems important. Consequently, when projects engage in highly regulated domains, they respect and internalize such specifications. Therefore, while the focus of policy-making may not be reflected in the regular governance concerns of developers, policies still act as a layer of fundamental governance that is seamlessly integrated into communities. Simply put, developers respect policies that are evidently important and extensively specified, but they are also faced with other concerns beyond those where ASFI largely institutes policies.

The ASFI's policies show relatively less attention to the technical aspects that constitute communities' most governed activities (issues/patches, artifacts, etc.), suggesting that the foundation defers to the discretion and objectives of developers on these subjects. The generally lower policy internalization along core development concerns may also be explained by the observation that the existing technical regulations are general recommendations rather than specific guidelines. Hence, we see considerable governed activity along some of these ('issues'/'patches'/'builds'), reflecting project efforts to coordinate fluid communities, channel their contributions, adapt to emerging technologies, and meet release targets.

RQ3 examines the association of self-governance and internalization of foundation policies with the objective success of projects (Table. 4). It is based on the implicit assumption that projects will perform governance and adopt policies in a manner that helps them attain their objective, which is to graduate from the Incubator. The Incubator assesses projects based on the diversification of the community and the capability to produce compliant software. Interestingly, governance behavior around the more highly regulated governance topics does not stand out as a significant discriminant between graduated and retired projects.

Foundation policies may play a role in furthering development, facilitating coordination, and fostering consensus among communities, as analyses showed positive associations between internalization of voting timelines and odds of graduation. We also find some evidence that community initiative in less regulated governance areas supports project sustainability. Projects that coordinate submission and incorporation of patches more often are both building their community and improving their product, making them more likely to graduate. Such projects were likely able to step up to the limited presence of technical guidelines or recommendations from the ASFI by instituting their own routines to sustain development.

We have one significant finding around a highly regulated topic: Incubator reporting. We found a negative association between levels of governed activity around Incubator reporting and the odds of graduation. Reporting to the Apache Incubator is intended to motivate project performance as well as track their progress [105]. Therefore, it is interesting that more formalization is associated with a reduced likelihood of graduation for a highly regulated topic. We further explain that this effect from Incubator reporting likely does not imply a straightforward causal relation between formalization and success. It often presents a delicate situation for already struggling projects, as they are compelled to focus on implementing formal policy. This has sometimes proven to be especially burdensome for small projects. Apache Gossip is such an example, where the small community struggled with the overhead of implementing the regular reporting protocols set by the ASFI and was eventually retired [37].

All in all, communities in the ASFI are bound by its requirements, especially in domains that elicit a greater volume of formalization. At the same time, their actual governance concentrates on aspects distinct from the ones in which ASFI regulates the most. Importantly, we find limited support for the argument that projects should embrace prevailing models of OSS formalization, be it in terms of aligning governance focus or internalizing policies in more regulated topics, to successfully realize their objectives. Therefore, written formal policies from OSS communities may not be a comprehensive account of how their actual governance unfolds.

8 CONTRIBUTIONS

8.1 Practitioner Recommendations

Some of the chief takeaways from this study, for practitioners in technology policy in general and OSS in particular, are the joint importance of community-level coordination and formal organizing by foundations. Our findings indicate a seeming incongruence between the extent of regulation and operational activities reflected by the volume of governance discourse across topics. At the same time, more regulation also sees increased internalization among projects, thus standardizing their operations. Therefore, the strength of the foundation's position on a policy can be effectively communicated to projects through policy documents. This should not be taken to imply that foundations should endeavor to maximize internalization through increases in policy length, as more internalization and standardization are not necessarily always good (and, more fundamentally, this work does not establish the direction of their relationship). Rather, our findings should encourage them to create more comprehensive policies on topics that most reflect their priorities.

Foundations may increase congruence between policy-making and operational priorities by actively soliciting feedback from mentored projects in designing their governance. They may benefit from examining the practical interpretations of formal policies to identify governance domains that may require more policy-making or even revise existing policies to serve mentored projects better. While ASFI/ASF strives towards democratic, community-first ideals, such participation may not yet be prevalent across all OSS support organizations. With a governance system informed by the low-level daily experiences of developers, a foundation enjoys the legitimacy of its membership while promoting sustained production of quality digital public goods.

Our results indicate a positive relationship between internalization of voting timelines and graduation rates. Projects call votes to ratify releases, add new committers, elect members of management committees, and graduate or retire from the Incubator. Voting is thus an integral mechanism among the consensus-driven democratic communities of the ASFI and high internalization of voting timelines means that projects strictly follow an established pattern for major processes. Projects are, hence, expected to be successful when they establish and strictly follow guidelines that mobilize participation in a timely order and facilitate distributed decision-making.

Project success is also related to the extent of governed activity in less formalized aspects, such as patch management. Communities are encouraged to set up self-governance mechanisms to coordinate contributions and incorporate appropriate corrections and enhancements. Such systematic measures are expected to improve product quality, user base, and overall project viability.

Periodic status reports are a key tool for both foundations and the projects within them. These reports allow foundations to ensure their projects are conforming with their policies and also provide an opportunity for projects to signal a need for additional mentorship or attention. Our results indicate a negative relationship between activity expended in reporting and graduation rates. Reporting introduces additional tasks for projects as they assign responsibilities, gather information across the community to fill in updates, and coordinate reviews through the mentors/designated committees (PPMC in the case of ASFI) and may be too costly for volunteer communities in terms of time and effort. Foundations may benefit from evaluating the efficacy of such reporting protocols as they are currently observed and suitably adjust the process (e.g., reporting frequency, accommodations for missed reports, etc.) to streamline developer efforts while also tracking their progress and offering them mentorship when it will be most beneficial.

8.2 Further Work

Our work opens up multiple research questions in pragmatic and principled governance design for socio-technical systems in general and OSS in particular. Future work in community-focused governance may explore established policies and discussions about them in foundations besides the ASFI, allowing us to generalize or understand nuances across different formal governance systems in OSS. This could also be expanded qualitatively. Systematic interviews with foundation-affiliated or independent project contributors on salient policy aspects can help probe causal connections between governance-related functions and project sustenance. Automated approaches in conversation-based behavioral analysis can be further augmented with the richness of qualitative insight. Language modeling-based NLP tasks may aid in the extraction of relevant developer exchanges, while focused analysis of the same by domain experts may reveal valuable insight into conditions and policies that see reinterpretation or divergence, and the manner or extent of such events. Such studies hold immense potential in advancing and informing practical initiatives in OSS governance.

Through our research questions, we uncover the full spectrum of themes represented by ASFI policies and the relative policy-making focus across them. Understanding why and how often policy is instituted for different aspects of community management is integral to informing governance design for projects at large. We encourage researchers to explore differences and similarities between different foundations in terms of the projects they support, the types of policies they enact, and the concerns they address.

Research on governance design can be extended beyond foundations to investigate the dynamics among cases of emergent, bottom-up formalization. Researchers may examine relationships between policy and operations among independent communities with documented governance systems (e.g., GOV.md policy files [103] or codes of conduct [48]) set up and run by developers instead of organizations. This further allows us to understand OSS governance and outcomes among the majority of projects running in the absence of centralized, foundation-driven mechanisms.

9 THREATS AND VALIDITY

Our findings may carry specific implications for Apache community members or the OSS ecosystem in general. However, we advise practitioners to exercise sufficient insight in interpreting our study and enacting policy reforms situated in the needs and objectives of their specific foundations/projects. This particularly applies to some effects in our findings, like Incubator reporting, which may not have a direct causal interpretation. Similarly, while our findings suggest that policy internalization around scheduled voting correlates with project graduation from the ASFI, directed focus on voting-related policy-making may not benefit all projects or foundations. Future replication across more organizations is hoped to enrich OSS governance research with more general insights.

For the purposes of our study, we treat ASFI's standards for graduation as an evaluation of OSS success and viability. The ASFI's stated objectives and standards provide a well-rounded criteria to assess the relation of governance behavior with viable and sustainable communities (Section. 4). It should be noted, however, that projects sometimes have varied reasons for choosing to graduate or discontinuing incubation, varied enough to cast doubt on simplistic associations of "graduation" with "success" and "retirement" with "failure". Reasons include but are not limited to their sense of cultural fit or need for ASF's specific portfolio of support servers. Therefore ASFI graduation, while considered a respected and tested model of evaluation, may not generalize to a conclusive metric of OSS success.

Our work is based on large public mailing lists. While these are the central communication channels in the ASFI, projects also maintain private lists reserved for certain more sensitive project businesses, including committer voting, etc. These are restricted from public access and are currently beyond our scope. ASFI leadership discourages the use of these lists as much as possible, and they are typically only used for "personnel" matters, such as if a contributor is breaking a project's code of conduct or to vote in new committers.

Our study rests on information extracted by semi-supervised learning. The choice of semi-supervised learning was largely motivated by our constructs (Section. 5.3), the limits of supervised learning, and, most importantly, to facilitate scalable organizational insight. Unsupervised/semi-supervised methods have known limitations and may require specialized approaches and metrics to validate. We tuned the performance of our clustering models with established measures such as clustering validity and NPMI-based [5] topic coherence (Appendix. B). Moreover, the very high values of R^2 that we report for our models are an encouraging sign that these constructs are credibly capturing important aspects of project governance activity.

We named the resulting topic clusters by examining the most frequently used words in them, activities, and the policies to which they were assigned. This qualitatively distills the essence of the clusters and makes it possible for us to interpret them for purposes of downstream analyses. Therefore, interpretations of topics and associated effects may vary across researchers. Through further checks, we find that the topics found in the main and supplementary analysis are largely even if not perfectly identified (Appendix. A.2).

While we used domain-adapted language models whenever available, some tasks like semantic role labeling and semantic similarity scoring were more specialized, with limited models and datasets available. Annotating training data consistent with benchmark datasets is complicated for such tasks and limits the scope of the methodology for replicating results. In such cases, we used models trained on standardized benchmark datasets [9, 97], and our subjective review of resulting measurements confirmed their performance as consistent with our constructs.

Certain project mailing lists did not reflect governed activity under all of the 42 different governance topics. This could be attributed to the extent of engagement or varied priorities across projects. For example, resource object management routines are likely exclusive to Java-based projects. Moreover, the cluster-size hyperparameter from HDBSCAN sets a lower threshold on how frequent an activity needs to be in order to be considered a cluster. While this captures the more prevalent developer practices, it may result in certain less frequent activities among projects being treated as outliers.

In section. 5.3, we explain the computing overheads and limits on input size for transformer-based language models. This often caused the exclusion of broader text context from social interactions [104]. We encountered a few cases in our dataset where extensive policies with multiple nested or bulleted conditions were truncated during intermediate preprocessing or parsing stages. Ongoing efforts at supporting longer context windows [90] for representation learning should expand the scope of language models for discourse analysis.

10 CONCLUSION

Open source software projects join foundations like the Apache Software Foundation despite the "anti-regulatory" tendency of many OSS developers. They do so because the standardized, streamlined governance systems that foundations operate provide clarity, best practices, mentorship, economies of scale, and lower administrative overhead. Yet OSS projects may simultaneously find themselves benefiting from formal structure and/or constrained by it to varying degrees.

While it is a widely accepted truism that governance in practice often differs from governance in form, it has been a challenge to demonstrate this at scale and determine the manner in which formal depictions and ground behavior diverge. Articulating fundamental questions about governance practices through NLP methods, particularly language modeling, enables us to quantify the governance behavior of projects, including how they govern themselves and internalize formal policy.

We find that while OSS communities are generally framed by formal regulations, they focus their practical governance efforts in a manner distant from the thrust of formal policy-making. Further, their governance behavior around highly formalized concerns seems to have little bearing on their rates of success and sustainability. What stands out is the adaptability of their governance efforts as well as their internalization of policy around relatively less regulated topics. In conclusion, a comprehensive understanding of peer production and other types of collective action must account for an institution's formal structure, actual governance practice, and the effective structures that emerge from their interaction, which we measure at scale.

ACKNOWLEDGMENTS

We would like to thank Anamika Sen, Ph.D. candidate in Economics at the University of Amherst, Massachusetts, and Likang Yin, Ph.D. candidate in Computer Science at the University of California, Davis, for contributing their data. The study was supported by the National Science Foundation under the Growing Convergence Research Scheme (Grant No #2020751).

REFERENCES

[1] Curtis Atkisson. 2022. Mentors matter: Association of mentors with project success in the Apache Software Foundation Incubator. *PLOS ONE* 17, 8 (Aug. 2022), e0272764. https://doi.org/10.1371/journal.pone.0272764

[2] John Langshaw Austin. 1975. *How to do things with words*. Vol. 88. Oxford University Press, Oxfork, UK.

[3] Yochai Benkler. 2006. The Wealth of Networks: How Social Production Transforms Markets and Freedom.

[4] Claire Bonial, Olga Babko-Malaya, Jinho D Choi, Jena Hwang, and Martha Palmer. 2010. Propbank annotation guidelines. *Center for Computational Language and Education Research, CU-Boulder* 9 (2010), 90 pages.

[5] Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL* 30 (2009), 31–40.

[6] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. arXiv:508.05326

[7] John Seely Brown and Paul Duguid. 1991. Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. *Organization science* 2, 1 (1991), 40–57.

[8] Brian Butler, Elisabeth Joyce, and Jacqueline Pike. 2008. Don't look now, but we've created a bureaucracy: the nature and roles of policies and rules in wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1101–1110. https://doi.org/10.1145/1357054.1357227

[9] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens (Eds.). Association for Computational Linguistics, Vancouver, Canada, 1–14. https://doi.org/10.18653/v1/S17-2001

[10] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Eduardo Blanco and Wei Lu (Eds.). Association for Computational Linguistics, Brussels, Belgium, 169–174. https://doi.org/10.18653/v1/D18-2029

[11] Michael D. Cohen, Roger Burkhart, Giovanni Dosi, Massimo Egidi, Luigi Marengo, Massimo Warglien, and Sidney Winter. 1996. Routines and other recurring action patterns of organizations: contemporary research issues. *Industrial and corporate change* 5, 3 (1996), 653–698.

[12] FLAX/JAX Community. 2021. MPNET Stack Overflow. Hugging Face. https://huggingface.co/flax-sentence-embeddings/stackoverflow_mpnet-base

[13] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116

[14] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 670–680. https://doi.org/10.18653/v1/D17-1070

[15] Sue ES Crawford and Elinor Ostrom. 1995. A grammar of institutions. *American political science review* 89, 3 (1995), 582–600.

[16] K Crowston, K Wei, Q Li, Ugur Eseryel, and J Howison. 2005. Coordination of Free/Libre Open Source Software Development. In *ICIS 2005*. International Conference on Information Systems, Las Vegas, NV, 14 pages.

[17] Richard Cyert and James March. 2015. Behavioral theory of the firm. In *Organizational Behavior 2*. Routledge, Oxfordshire, England, UK, 60–77.

[18] Paul B De Laat. 2007. Governance of open source software: state of the art. *Journal of Management & Governance* 11 (2007), 165–177.

[19] Leisha DeHart-Davis, Jie Chen, and Todd D. Little. 2013. Written Versus Unwritten Rules: The Role of Rule Formalization in Green Tape. *International Public Management Journal* 16, 3 (Jul 2013), 331–356. https://doi.org/10.1080/10967494.2013.825193

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[21] Paul J. DiMaggio and Walter W. Powell. 1983. The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review* 48, 2 (1983), 147–160.

[22] John Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis* Special volume of the Philological Society (1957), 10–32.

[23] The Apache Software Foundation. 2023. Guide to successful graduation. https://incubator.apache.org/guides/graduation.html

[24] Jo Freeman. 1972. The Tyranny of Structurelessness. *Berkeley Journal of Sociology* 17 (Jan 1972), 151–164. http://www.jstor.org/stable/41035187?seq=1#page_scan_tab_contents

[25] Seth Frey and Robert W Sumner. 2019. Emergence of integrated institutions in a large population of self-governing communities. *PloS one* 14, 7 (2019), e0216335.

[26] Seth Frey, Qiankun Zhong, Beril Bulat, William D Weisman, Caitlyn Liu, Stephen Fujimoto, Hannah Wang, and Charles M Schweik. 2022. Governing online goods: Maturity and formalization in Minecraft, Reddit, and World of Warcraft communities. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–23.

[27] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, Eunjeong L. Park, Masato Hagiwara, Dmitrijs Milajevs, and Liling Tan (Eds.). Association for Computational Linguistics, Melbourne, Australia, 1–6. https://doi.org/10.18653/v1/W18-2501

[28] R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 1–28. https://doi.org/10.1145/3449249

[29] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv:2203.05794

[30] Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, 364–369. https://doi.org/10.18653/v1/P18-2058

[31] Robert Heckman, Kevin Crowston, U Yeliz Eseryel, James Howison, Eileen Allen, and Qing Li. 2007. Emergent decision-making practices in free/libre open source software (FLOSS) development teams. In *Open Source Development, Adoption and Innovation: IFIP Working Group 2.13 on Open Source Software, June 11–14, 2007, Limerick, Ireland 3*. Springer, New York, USA, 71–84.

[32] Robert Heckman, Kevin Crowston, U. Yeliz Eseryel, James Howison, Eileen Allen, and Qing Li. 2007. Emergent Decision-Making Practices in Free/Libre Open Source Software (Floss) Development Teams. In *Open Source Development, Adoption and Innovation*, Joseph Feller, Brian Fitzgerald, Walt Scacchi, and Alberto Sillitti (Eds.). Vol. 234. Springer US, Boston, MA, 71–84. https://doi.org/10.1007/978-0-387-72486-7_6 Series Title: IFIP — The International Federation for Information Processing.

[33] Jérôme Hergueux and Samuel Kessler. 2022. Follow the Leader: Technical and Inspirational Leadership in Open Source Software. In *CHI Conference on*

Human Factors in Computing Systems. ACM, New Orleans LA USA, 1–15. https://doi.org/10.1145/3491102.3517516
[34] Francis Heylighen. 2013. Self-organization in Communicating Groups: the emergence of coordination, shared references and collective intelligence. In *Complexity perspectives on language, communication and society*. Springer, New York, USA, 117–149.
[35] Eric von Hippel and Georg von Krogh. 2003. Open source software and the "private-collective" innovation model: Issues for organization science. *Organization science* 14, 2 (2003), 209–223.
[36] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*. ICLR, New Orleans, LA, 14 pages. https://openreview.net/forum?id=SkxgnnNFvH
[37] Apache Software Foundation Incubator. 2018. [DISCUSS] Leave apache and the incubator. https://lists.apache.org/thread/m960h2soj314ty6yvr83nvsp26lln9jb
[38] Javier Luis Cánovas Izquierdo and Jordi Cabot. 2018. The role of foundations in open source projects. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Society*. Association for Computing Machinery, New York, NY, 3–12.
[39] Javier Luis Cánovas Izquierdo and Jordi Cabot. 2020. A Survey of Software Foundations in Open Source. http://arxiv.org/abs/2005.10063
[40] Jack Jamieson, Eureka Foong, and Naomi Yamashita. 2022. Maintaining Values: Navigating Diverse Perspectives in Value-Charged Discussions in Open Source Development. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), 1–28. https://doi.org/10.1145/3555550
[41] Chris Jensen and Walt Scacchi. 2010. Governance in open source software development projects: A comparative multi-level analysis. In *IFIP International Conference on Open Source Systems*. Springer, New York, USA, 130–142.
[42] Shagun Jhaver, Seth Frey, and Amy X Zhang. 2023. Decentralizing Platform Power: A Design Space of Multi-level Governance in Online Social Platforms. *Social Media+ Society* 9, 4 (2023), 20563051231207857.
[43] Daniel Jurafsky and James H Martin. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.
[44] John C. Lammers. 2011. How institutions communicate: Institutional messages, institutional logics, and organizational communication. *Management Communication Quarterly* 25, 1 (2011), 154–182.
[45] Jean Lave and Etienne Wenger. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge University Press, Cambridge, UK.
[46] Gwendolyn K. Lee and Robert E. Cole. 2003. From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization science* 14, 6 (2003), 633–649.
[47] Barbara Levitt and James G March. 1988. Organizational learning. *Annual review of sociology* 14, 1 (1988), 319–338.
[48] Renee Li, Pavitthra Pandurangan, Hana Frluckaj, and Laura Dabbish. 2021. Code of conduct conversations in open source software projects on github. *Proceedings of the ACM on Human-computer Interaction* 5, CSCW1 (2021), 1–31. Publisher: ACM New York, NY, USA.
[49] Siegwart Lindenberg. 2003. Governance seen from a framing point of view: the employment relationship and relational signalling. In *The Trust Process in Organizations*. Edward Elgar Publishing, Cheltenham, UK, Chapter 3, 37–57. https://EconPapers.repec.org/RePEc:elg:eechap:2811_3
[50] Siegwart Lindenberg and Linda Steg. 2013. Goal-framing theory and norm-guided environmental behavior. In *Encouraging sustainable behavior*. Psychology Press, New York, NY, 37–54.
[51] JG March and HA Simon. 1958. *Organizations*. Wiley, New York.
[52] M. Lynne Markus. 2007. The governance of free/open source software projects: monolithic, multidimensional, or configurational? *Journal of Management & Governance* 11, 2 (Jun 2007), 151–163. https://doi.org/10.1007/s10997-007-9021-x
[53] Michael Dean McGinnis. 1999. *Polycentricity and local public economies: Readings from the workshop in political theory and policy analysis*. University of Michigan Press, Ann Arbor, MI.
[54] Michael D. McGinnis. 2011. An Introduction to IAD and the Language of the Ostrom Workshop: A Simple Guide to a Complex Framework: McGinnis: IAD Guide. *Policy Studies Journal* 39, 1 (Feb 2011), 169–183. https://doi.org/10.1111/j.1541-0072.2010.00401.x
[55] Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* 2, 11 (2017), 205.
[56] John W Meyer and Brian Rowan. 1977. Institutionalized organizations: Formal structure as myth and ceremony. *American Journal of sociology* 83, 2 (1977), 340–363.
[57] Vishal Midha and Anol Bhattacherjee. 2012. Governance practices and software maintenance: A study of open source projects. *Decision Support Systems* 54, 1 (2012), 23–32.
[58] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781
[59] Audris Mockus, Roy T. Fielding, and James D. Herbsleb. 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 11, 3 (2002), 309–346.
[60] Davoud Moulavi, Pablo A Jaskowiak, Ricardo JGB Campello, Arthur Zimek, and Jorg Sander. 2014. Density-based clustering validation. In *14th SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics Publications, Philadelphia, PA, 839–847.
[61] Richard R Nelson. 1985. *An evolutionary theory of economic change*. Harvard University Press, Cambridge, MA.
[62] Siobhan O'Mahony. 2005. Nonprofit Foundations and Their Role in Community-Firm Software Collaboration. In *Perspectives on Free and Open Source Software*. The MIT Press, Cambridge,MA. https://doi.org/10.7551/mitpress/5326.003.0028 arXiv:https://direct.mit.edu/book/chapter-pdf/2297482/9780262256124_cat.pdf
[63] Siobhán O'mahony and Fabrizio Ferraro. 2007. The emergence of governance in an open source community. *Academy of Management Journal* 50, 5 (2007), 1079–1106.
[64] Julian E Orr. 1990. *Talking about machines: An ethnography of a modern job*. Cornell University Press, Ithaca, NY.
[65] Elinor Ostrom. 2009. *Understanding institutional diversity*. Princeton University Press, Princeton, NJ.
[66] Elinor Ostrom, Marco A Janssen, and John M Anderies. 2007. Going beyond panaceas. *Proceedings of the National Academy of Sciences* 104, 39 (2007), 15176–15178.
[67] Siobhán O'Mahony. 2003. Guarding the commons: how community managed software projects protect their work. *Research policy* 32, 7 (2003), 1179–1198.
[68] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162
[69] Brian T. Pentland. 1995. Grammatical Models of Organizational Processes. *Organization Science* 6, 5 (Oct 1995), 541–556. https://doi.org/10.1287/orsc.6.5.541
[70] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Online, 101–108.
[71] Huilian Sophie Qiu, Anna Lieb, Jennifer Chou, Megan Carneal, Jasmine Mok, Emily Amspoker, Bogdan Vasilescu, and Laura Dabbish. 2023. Climate Coach: A Dashboard for Open-Source Maintainers to Overview Community Dynamics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–18. https://doi.org/10.1145/3544548.3581317
[72] Eric Raymond. 1999. The cathedral and the bazaar. *Knowledge, Technology & Policy* 12, 3 (Sep 1999), 23–49. https://doi.org/10.1007/s12130-999-1026-0
[73] Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* 3, 2 (2011), 2.
[74] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3980–3990. https://doi.org/10.18653/v1/D19-1410
[75] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, Shanghai China, 399–408. https://doi.org/10.1145/2684822.2685324
[76] Mohanbir Sawhney and Emanuela Prandelli. 2000. Communities of creation: managing distributed innovation in turbulent markets. *California management review* 42, 4 (2000), 24–54.
[77] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press, Cambridge, UK.
[78] Charles M Schweik and Robert English. 2007. Tragedy of the FOSS commons? Investigating the institutional designs of free/libre and open source software projects. *First Monday* 12 (2007), 35 pages. https://doi.org/10.5210/fm.v12i2.1619
[79] Charles M. Schweik and Robert English. 2013. Preliminary steps toward a general theory of internet-based collective-action in digital information commons: Findings from a study of open source software projects. *International Journal of the Commons* 7, 2 (Aug 2013), 234. https://doi.org/10.18352/ijc.397
[80] Charles M. Schweik and Robert C. English. 2012. *Internet success: a study of open-source software commons*. MIT Press, Cambridge, MA.
[81] John R Searle, Ferenc Kiefer, Manfred Bierwisch, et al. 1980. *Speech act theory and pragmatics*. Vol. 10. Springer, New York.
[82] Anamika Sen, Curtis Atkisson, and Charlie Schweik. 2022. Cui Bono: Do open source software incubator policies and procedures benefit the projects or the incubator? *International Journal of the Commons* 16, 1 (2022), 64–77.

[83] Sonali K. Shah. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. *Management science* 52, 7 (2006), 1000–1014.
[84] Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. arXiv:1904.05255
[85] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90, 2 (2000), 227–244.
[86] Saba Siddiki, Tanya Heikkila, Christopher M. Weible, Raul Pacheco-Vega, David Carter, Cali Curley, Aaron Deslatte, and Abby Bennett. 2022. Institutional analysis with the institutional grammar. *Policy Studies Journal* 50, 2 (2022), 315–339.
[87] Sameer B. Srivastava, Amir Goldberg, V. Govind Manian, and Christopher Potts. 2018. Enculturation Trajectories: Language, Cultural Adaptation, and Individual Outcomes in Organizations. *Management Science* 64, 3 (March 2018), 1348–1364. https://doi.org/10.1287/mnsc.2016.2671
[88] David Strang and John W Meyer. 1993. Institutional conditions for diffusion. *Theory and Society* 22 (1993), 487–511.
[89] Ştefan Stănciulescu, Likang Yin, and Vladimir Filkov. 2022. 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*. ACM, Singapore, 495–506.
[90] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long Range Arena : A Benchmark for Efficient Transformers. In *9th International Conference on Learning Representations*. ICLR, Virtual Event, 19 pages. https://openreview.net/forum?id=qVyeW-grC2k
[91] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 296–310. https://doi.org/10.18653/v1/2021.naacl-main.28
[92] Sentence Transformers. 2021. Cross-Encoder STSb-distilroberta-base. https://huggingface.co/cross-encoder/stsb-distilroberta-base
[93] Stef Van Buuren and Karin Groothuis-Oudshoorn. 2011. mice: Multivariate imputation by chained equations in R. *Journal of statistical software* 45 (2011), 1–67.
[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017), 11 pages.
[95] Karl E Weick. 1976. Educational organizations as loosely coupled systems. *Administrative Science Quarterly* 21 (1976), 1–19. https://doi.org/10.2307/2391875
[96] Karl E Weick and Karlene H Roberts. 1993. Collective mind in organizations: Heedful interrelating on flight decks. *Administrative Science Quarterly* 38 (1993), 357–381.
[97] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA* 23 (2013), 170.
[98] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. arXiv:1511.08198
[99] John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A Bilingual Generative Transformer for Semantic Sentence Embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 1581–1594. https://doi.org/10.18653/v1/2020.emnlp-main.122
[100] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. http://aclweb.org/anthology/N18-1101
[101] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv:1910.03771
[102] Xiaobao Wu, Chunping Li, Yan Zhu, and Yishu Miao. 2020. Short Text Topic Modeling with Topic Distribution Quantization and Negative Sampling Decoder. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 1772–1782. https://doi.org/10.18653/v1/2020.emnlp-main.138
[103] Yibo Yan, Seth Frey, Amy Zhang, Vladimir Filkov, and Likang Yin. 2023. GitHub OSS Governance File Dataset. arXiv:2304.00460
[104] Likang Yin, Mahasweta Chakraborti, Yibo Yan, Charles Schweik, Seth Frey, and Vladimir Filkov. 2022. Open Source Software Sustainability: Combining Institutional Analysis and Socio-Technical Networks. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–23.
[105] Likang Yin, Zhuangzhi Chen, Qi Xuan, and Vladimir Filkov. 2021. 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*. Association for Computing Machinery, Virtual, 1056–1067.
[106] Likang Yin, Zhiyuan Zhang, Qi Xuan, and Vladimir Filkov. 2021. Apache software foundation incubator project sustainability dataset. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. IEEE, Madrid, Spain, 595–599. https://doi.org/10.1109/MSR52588.2021.00081

A SUPPLEMENTARY ANALYSIS

A.1 Policies excluded from Primary Analysis

The supplementary analysis looks at community and foundation interaction over governance concerns and policies applicable to active incubation and mentoring. We remove certain categories of policy documents based on subsection headings in the original dataset [82]. These include "Steps to Retirement", "Deciding to Retire", 'Graduation discussion', "Graduation Approval vote", " The Graduation Process", "Preparing a Charter", "The Recommendation Vote", "Submission of Resolution to the Board", "Community Graduation Vote", "Press Releases for new Top Level Projects (TLP)", "Whether to graduate to Subproject or to top-level project", "Graduating to a Subproject", "ASF Board Resolution", "Post-Graduation tasks", "Transfering trademarks to the ASF" and "Subproject Acceptance vote". The aforelisted sections span terminal formalities and procedures to initiate and garner community/ASFI approval for graduation as well as steps towards formal induction into Apache. ASFI projects may pursue two modes of post-graduation affiliation: to function as a full-fledged independent top-level project (TLP) or as a subproject under a TLP. Sections also cover protocols to be observed when a project is being retired. All in all, 34 entries were removed out of the original 234. However, we retain policies that state the goals of the ASFI, expected standards, evaluation criteria, and other requirements meant to guide and mentor projects toward success.

A.2 Analysis and Results

We repeated all the steps through Section. 5.3 for detecting governed activities and measuring internalization with respect to the modified dataset. For RQ1, the correlation between the distributions of policy extent and governed activity was found to be 0.18 ($p = 0.41$). For RQ2, the correlation between the distribution of policy extent and the mean internalization by topic was found to be 0.69 ($p < 0.001$). These findings are nearly identical to those reported in the main text. The analyses for RQ3 are as below (see Table. 6). The differences between these findings are those reported in the main text and are discussed in the main text (Section. 6).

With the multitude of choices and considerations driving our study, our research questions presented two alternate analytic approaches to establishing our findings. As we draw parallels between the topical effects from the two analyses, we further assess how policy subsets driving them determine governance topics.

Rules act as initializing seeds in semantic clustering, while their words are used to reweigh cTF-IDF and detect related themes among governed activities. Graduation/Retirement protocols are closely

	Covariates Only		Covariates & Governed Activity		Covariates & Internalization		All Variables	
Predictor	Coefficient	p	Coefficient	p	Coefficient	p	Coefficient	p
Intercept	2.490	0.000	3.186	0.000	2.708	0.000	3.353	0.000
Committers[2]	0.077	0.874	-0.027	0.960	0.127	0.819	0.1745	0.766
Commits[2]	0.705	0.140	0.449	0.408	0.209	0.705	-0.0022	0.997
Developer Emails[2]	0.807	**0.016**	0.714	0.114	0.952	**0.021**	1.048	0.063
Incubation time[1]	-0.518	**0.011**	-0.358	0.286	-0.468	**0.044**	-0.238	0.515
Incubator Reporting[2]			-1.707	**0.001**			-1.584	**0.004**
Patches[2]			0.797	**0.010**			0.766	**0.030**
Voting Protocol/Timeline[2]			1.012	**0.002**			0.6686	0.070
Voting Protocol/Timeline[1]					0.761	**0.004**	0.800	**0.013**
Community[1]					-0.902	**0.000**	-0.5161	0.052
Observations: 208	R^2 (Tjur): 0.258 AIC: 139.91		R^2 (Tjur): 0.442 AIC: 115.36		R^2 (Tjur): 0.413 AIC: 123.49		R^2 (Tjur): 0.508 AIC: 109.76	

[1] Standardized variables [2] Log transformed (base 10) and standardized variable

Table 6: Summary: Binomial (Logit) GLM regression of project governance against Graduation/Retirement

associated with multiple other aspects of ASFI governance. E.g., the reduced dataset has fewer policies about the ASF board and their meetings and resolutions. These functions and offices are mostly observed when projects apply for graduation. As a result, removing a subgroup of policies redistributes the relative thematic and semantic salience of seeds, optimal training hyperparameters, and how the resulting pipeline clusters activities related to policies or identify topics.

The 200 policies in the supplementary analysis were parsed into 328 rules (Section. 5.3.1), which guided clustering and topic modeling. Out of the 154 topics discovered, 24 were found to span both community-governed activities and ASFI policies. Topic names were inferred in a manner similar to what we did for the primary analysis, i.e., based on policies the topics contained, their top keywords, and sampled governed activities. The retrained pipeline assigned topics to 219 of these 328 rules, leaving 109 outliers. Policy documents often contain canonical descriptions of norms and processes that may be dated and removed from practice [7, 64]. 57 outliers were common between the primary and the supplementary analysis. These included licensing terms unlikely to be frequent in conversations or recommendations and best practices, which may see relatively less translation into operations when compared to binding policies (e.g., "3 mailing list moderators are recommended").

Despite these differences, we found considerable consistency between topics and findings from the primary and supplementary analyses. 22 of the 24 topics in the supplementary analysis largely corresponded to topics from our primary analysis, which we used to interpret effects across the two analyses. This was confirmed further by an average match of 82.5% between the top 3 representative words from these topics and their counterparts from the primary analysis. The two extra topics, 'Project Webpage' and 'Request Handling', also overlap with URL/Links and General Communications, respectively, in terms of policies and types of activities they contained (See Appendix. C). Apart from some instances of reassignment across topics, 89.5% of all the 219 non-outlier rules in the supplementary analysis belonged to the same topics as they did during the primary analysis.

The primary and supplementary analyses preserve results from RQ1 and RQ2. One of the policies that contributed to an effect ('Project Configuration') in the primary analysis was an outlier and was not included among topics in the supplementary analysis. We discuss our interpretations and recommendations from RQ3 only based on effects that are supported by both independent experiments.

B SEMI-SUPERVISED PIPELINE: TRAINING AND VALIDATION

We leverage advances in natural language learning to train a pipeline to support behavioral research in online governance across multiple platforms, services, and communities of interest. In Section. 5.3.2, we discuss in detail our motivations for pursuing semi-supervised methods. We hereby provide further details on the choices we made in training and automated metrics selected for validation. For the purposes of reproduction, it is recommended that experiments are performed using an NVIDIA T4 GPU.

B.1 Sentence Embeddings and Clustering

The bi-encoder architecture was developed for computationally efficient semantic encoding of texts [74]. They involve training a Siamese network of two identical transformers to generate contextual encodings for two distinct text inputs. The averaged output from each transformer is then subjected to a contrastive loss objective function. By the end of the joint fine-tuning, both the transformers are capable of independently generating semantic embeddings for any given text input.

Huggingface [101] hosts multiple domain-specific bi-encoders. To generate semantic representations for all text data, we selected a model capable of interpreting a range of developer discussions. We use a bi-encoder trained on an extensive corpus of 18.5 million posts from Stack Overflow [12], a question-answering platform frequented by software engineers for discussions across all pertinent aspects of development, including licensing. We use the embeddings thus produced to cluster related activities.

For best clustering results, we conducted hyperparameter tuning for BERTopic's HDBSCAN through density-based clustering validity (DBCV) [60] over a sample of 100,000 activities. DBCV scores rate density-based models from -1 to +1, with higher values indicating better clustering quality. Maximizing DBCV ensures that high-density clusters are clearly discernible and separated by low-density regions in the semantic embedding space. This mitigates the misassignment of less related activities to clusters and also improves the overall consistency of subsequent topic assignments. To find optimal hyperparameters, we tune over the following HDBSCAN arguments: minimum cluster size and minimum samples. Higher values of cluster size threshold might lead to the merging of clusters, while greater sample size promotes dense clustering and more outliers. Both parameters were varied in combinations from 10 (0.01% of sample size) to 100 (0.1% of sample size). Prior to clustering, BERTopic also uses Uniform Manifold Approximation and Projection or UMAP for dimension reduction of embeddings. The number of neighbors parameter in UMAP decides the trade-off between preserving the global and local structure and was also varied between 10 and 100. We retain the model with the best relative DBCV score at 0.32. For the supplementary analysis, we appropriately re-conduct hyperparameter search over the same ranges to a DBCV score of 0.33.

B.2 Topic Modeling

We infer interpretable topic names from clusters through their top 3 representative words generated by cTF-IDF [29], policies associated, and governed activities sampled from the same. For evaluating the quality of the topic representative words, we chose the C_v metric [75]. For a given set of representative words, C_v measures how coherent these are with the content of the documents (or clusters) they represent. Among several topic modeling metrics, C_v scores showed the highest correlation with human ranking of representative topic words for multiple datasets [75]. It calculates probabilities of co-occurrence [5] for each representative word with other words from the content being modeled and returns an aggregate measure for the set on a scale of 0 to 1. Wu et al. [102] provide a detailed breakdown of the algorithm used to calculate C_v, where values close to 1 are generally attained for ideal hypothetical cases of uniform data when texts comprise identical groups of words (N-grams). Our models for the primary and supplementary analysis attain C_v scores of 0.683 and 0.695, respectively.

The authors intermittently read and reviewed the data and results from the pipeline. We took this approach to human validation rather than scaling to crowd-workers, primarily because of the domain expertise in open source, software development, and related policies required to categorize and evaluate statements in our corpus. We found general consistency between the representative words and their presence in their respective topic content. In several cases, we found synonyms or other words closely associated with the representative words. This is expected and can be attributed back to the clustering segment of our pipeline, which works on the principle of distributional semantics [22] (i.e., the meanings of words are qualified by the context they are situated in) to identify related activities. Examples include 'Zoom' and 'Graphics,' 'Volunteer' and 'Help' or 'Configuration' and 'Infrastructure.' Appendix. C provides sampled examples across all the topics in our primary analysis.

C TOPIC APPENDIX

The following tables list the 42 governance topics we identified across emails and ASFI policies in the primary analysis. In order to inform subjective interpretation, we include a brief description of each topic, their top 3 representative words from BERTopic's cTF-IDF (Section. 5.3), a sampled governed activity, its closest related ASFI policy (Section. 5.3.4), and the internalization between the two. Figure. 3 provides a visualization of policy internalization scores within different governance topics. Topics that appeared in both the primary and supplementary analyses are indicated by (*).

Topic	Top Words	Description	Policy	Governed Activity	Internalization
Apache Foundation/ Contributor License*	ASF, ICLA, Projects	Policies and discussions applicable across the Apache Foundation, including contributor licenses and post-development release compliance	"There are restrictions about how projects can use their own and other ASF brands and names"	"The ASF has strict policies on what can be used by ASF projects to ensure that the final distribution is compatible with the Apache license" (Derby, 2005-04-29)	0.55
Graduation Requirements/ Maturity Model*	Graduation, TLP[1], Address	Conditions, standards of assessment and recommendations for successful graduation from the ASFI	"Once the project starts feeling ready to graduate they should make a self-assessment of that readiness"	"I indeed believe that you could graduate any time now – once you have some experience in adding to the community" (Isis, 2011-12-11)	0.58
Email Communications*	Email, Mail, Mailing	Policies as well as best practices encouraging use of ASFI project mailing lists	"Note that subscribers of external mailing lists will not be automatically subscribed to the new incubator project mailing lists"	"I thought the dev@ and user@ lists were normally discarded if not sent from a subscriber" (Slider, 2014-05-15)	0.42
Podling Project Management Committee (PPMC)*	PPMC, PMC, Members	Constitution, responsibilities and scope of authority of the management committee (PPMC) assigned to an individual incubating project	"Discussions of PPMC candidates and votes happen on the project's private PPMC mailing list"	"Of course we would immediately set discussing nominees drawn from the list of PPMC members as it exists today" (Vxquery, 2013-11-19)	0.55
Incubator Reporting*	Report, Reports, Month	Steps for projects to prepare status/progress reports for the ASFI, and associated activities	"Mentors must sign off on project reports"	"(report signed - off.)" (Amaterasu, 2017-10-08)	0.67
Processes	Process, Shutdown, Run	Particular processes related to development and code management	"A specific process exists for donating code to the ASF which projects need to follow"	"Once I am able to I'll run some tests" (Storm, 2014-08-07)	0.01
Committers/ Commits*	Commit, Committers, Committer	Practices and norms among ASFI affiliated committers, including developer admissions and their duties.	"As a project grows it needs to renew itself by accepting new committers"	"We still looking to grow the community and add new committers" (Shindig, 2009-03-10)	0.56
Community*	Community, Open, Source	Expectations and cultural codes observed among Apache communities	"Before a project graduates it must create a diverse and self - sustaining community"	"we need to create a healthy community" (Gearpump, 2016-04-05)	0.62
Project/ Developers	Project, Projects, Developers	General norms and discussions related to ASFI projects and developers	"Top level projects[1] are created by a resolution by the board"	"This is an active project" (Falcon, 2013-09-05)	0.37

Topic	Top Words	Description	Policy	Governed Activity	Internalization
Incubator Project Management Committee (IPMC)*	IPMC, Votes, Vote	Constitution, responsibilities and scope of authority of the management committee (IPMC) overseeing the Incubator program	"For a project to receive full permission from the IPMC to execute the release the release vote must be held on the incubator general list"	"Yes ask the IPMC to ratify the 1 . 3 . 0 - incubating - alpha release" (Wicket, 2007-03-24)	0.53
Links/URLs	Link, Links, URL	Adding, updating, connecting and preserving weblinks and associated content	"A non-ASF release may not be linked from a project's website"	"Fixed the Youtube link" (Kudu, 2016-04-28)	0.13
General Communications	Request, Response, Reply	Communications beyond mailing lists, such as sending and receiving data and resources from admins, servers etc.	"Some resources are created by infrastructure after an appropriate request"	"I also agree that request cycle should be responsible for detaching" (Wicket, 2007-03-25)	0.25
Documentation	Doc, Document, Docs	Creating, using and updating appropriate documentation for product/tool manuals, as well as project/ASFI related content	"The original proposal and the status document should be consulted when creating the graduation resolution document"	"GCV5 has no document associated, since it's too early and unstable for serious document efforts" (Harmony, 2006-09-11)	0.28
Tasks Handling	Task, Tasks, Job	Creation, navigation and successful management of tasks, at both system and project level	"When retiring open a 'task' INFRA JIRA ticket 'Retire the $Project Incubator Project', open sub-tickets using 'create sub-task' as applicable"	"When the next iteration of the subDAG would run, it could not execute the subDAG tasks since they were set to failed" (Airflow, 2017-11-17)	0.36
Project Directory	File, Files, Directory	Rules and practices around files and directory management among projects	"When retiring, create a file retired.txt at the top-level of each project's source repository"	"I setup a putfile processor with the local directory of machine B" (Nifi, 2015-04-28)	0.34
JIRA*	JIRA, Browse, JIRAs	Activities involving JIRA, an issue/task tracking utility	"Resources should be requested from Infra via Apache Infra JIRA → create task"	"Given a general feeling that if we could use Spnego authentication, we can file a JIRA to add it." (Knox, 2013-08-29)	0.41
Visibility/ Resolution	Zoom, Look, Good	Activities related to front-end in product and ASFI/project related web resources	"When retiring modify the resolution attributes of the podlings.xml appropriately"	"You probably should not listen to me when it comes to graphics since I'm really bad at visual design" (OpenOffice, 2012-02-14)	0.15
Deletion/ Removal*	Delete, Remove, Removed	Removal of redundant information and artifacts from checklists, trackers, code base, storage etc.	"The project startup template contains a list of actions, all those which do not apply should deleted"	"The caller is responsible for cleaning up module." (Impala, 2016-05-25)	0.05

Topic	Top Words	Description	Policy	Governed Activity	Internalization
Security/ Authentication	Security, User, Password	Aspects related to secure development and products, such as authentication and privacy preservation	"When retiring move the $project JIRA to 'retired' and set to read - only"	"A regular user will create the notebook and change the permission settings to his/her notebook" (Zeppelin, 2016-02-06)	0.1
Issues*	Issue, Fix, Problem	Norms and activities articulating, verifying, and resolving pertinent issues around development and release.	"By the time you graduate all issues listed in the disclaimer must be corrected"	"I'm not sure how to fix this with Wagon" (Ariatosca, 2017-08-16)	0.04
Incubation Process	Incubator, Incubation, Incubating	Actions related to Incubation, particularly graduation/retirement	"The final decision to retire the project takes the form of a vote by the IPMC on general@incubator"	"Pirk would just retire from incubation." (Pirk, 2017-02-26)	0.52
Mailing list/ Whimsy[2]	List, Lists, Whimsy	Rules and practices regarding the general Incubator mailing list	"IPMC members are on the general list, so posting to the general list is sufficient."	"I'll send a note to the list when I get further along." (Qpid, 2008-04-21)	0.30
Software Installations	Install, Installation, Ubuntu	Installation of utilities and dependencies as part of specific development, or provided by the ASFI	"Incubation proposal should include a list of required resources which will require active set up"	"If you need sudo you need to set up Orthrus (Opie) first" (Bloodhound, 2013-02-11)	0.22
Forms	Form, Forms, Onsubmit	Paperwork, templates and web forms related to ASFI as well as product in development	"The nominating PPMC member should send a message to the IPMC with a reference to the vote result in the following form:"[3]	"a problem with nested forms arises when the outer form is submitted" (Wicket, 2006-11-05)	0.25
Proposals/ Resolutions	Proposal, Solution, Task	Formulation of plans and objectives among developers, as well as sharing of roadmaps and solutions	"A suitable board resolution should be drawn by the community advised by the mentors"	"Comments to my earlier proposal are welcome" (Netbeans, 2016-10-06)	0.13
Patches*	Patch, Patches, Attached	Applying, reviewing, committing and all related discussion of patches	"When they give a good answer that the documentation doesn't cover ask developers/users to submit a patch"	"[NAME] has uploaded a new patch set (#2)." (Impala, 2016-03-09)	0.28
Checks/Tests	Check, Checked, Checks	Priority activities which need to be completed, validated, and confirmed	"add project to the Incubator's content/podlings.xml as soon as possible after acceptance."	"Having the checks in code makes us search for the problems in the right place" (Corinthia, 2015-02-19)	0.07
General Voting*	Vote, Votes, Voting	Recommended and required voting related protocols across ASFI projects	"If the release vote passes the project must send a summary of that vote to the Incubator's general list"	"You must indicate which measure you are voting on in order for your vote to be counted" (Hcatalog, 2013-01-28)	0.43

Topic	Top Words	Description	Policy	Governed Activity	Internalization
Copyright Notice*	License, Notice, Copyright	Adoption, review and distribution of appropriate license copyrights and notices	"When retiring, the copyright checkbox of the project's incubation status page has to be checked off."	"I don't believe the return values can be copyrighted" (Harmony, 2006-07-01)	0.29
Schedules/ Events*	Day, Date, Week	General timed activities among developers	"Post report on the month-year page with the provided template"[4]	"PR96 - owned by [NAME], he will be back after new year" (Trafodion, 2015-12-23)	0.03
Release Management*	Release, Releases, Version	Protocols and activities related to developing releases (ASF and non-ASF)	"Anybody reviewing your releases will explain what they checked and what they found"	"Help answer any question and check releases." (Apisix, 2020-06-03)	0.67
Voting Protocol/ Timeline*	Hours, Vote, Open	Rules and activities concerning timing/duration of ASFI-related voting and associated processes.	"The board[6] vote is effective immediately although the public minutes of the board meeting appear later, usually within a month"	"I'll run the voting thread tomorrow afternoon/evening" (JSPwiki, 2012-06-05)	0.32
Project Configuration*	Config, Configure, Configuration	Operations related to configuring settings and parameters across software/hardware	"Incoming project needs to set up its infrastructure."	"Shouldn't it let this value be set by cloud-setup-agent or by the admin manually?" (Cloudstack, 2012-08-08)	0.41
Release Blockers	Blocker, Blockers, Wait	Serious technical faults (e.g., code issues) and associated organizational roadblocks (e.g., failed release votes) which may suspend/delay releases	"Some issues may be blockers"	"The sad thing is that we have 4 blocker and 19 critical issues with no responsible persons" (Netbeans, 2018-04-20)	0.49
Builds	Build, Builds, Built	Activities around creating, maintaining and deploying builds	"When retiring turn off $ project automatic builds"	"You can't do that until Geronimo is build once successfully" (Geronimo, 2004-02-25)	0.20
Project Wiki	Wiki, Page, Mediawiki	Rules and activities around establishing and maintaining project wikis	"When retiring make $project wiki read - only"	"The wiki is in need of TLC" (Cloudstack, 2012-11-01)	0.26
Apache Incubator*	Apache, Incubator, Dubbo[6]	Rules and restrictions generally applicable across newly incubating projects, and associated discussions	"Do not request source repositories before SGAs are filed for instance, if the source code is not already Apache licensed or category A licensed"	"We strongly recommend that the first Apache release is source only" (Druid, 2018-07-11)	0.43

Topic	Top Words	Description	Policy	Governed Activity	Internalization
Resource Object Management*	Resource, Resources, Resourcetype	Operations involving hardware and software resources among projects	"Apache Infra has a guide that you can use to understand the flow of resource requests"	"Diverting from the 'normal' way of doing this with resource files without having a problem at the moment feels like a premature optimization to me" (Harmony, 2006-07-13)	0.26
Subversion (SVN)*	SVN, Subversion, Diff	Activities involving Subversion, a VCS system popular in ASFI	"Committers can access the project template for resolutions in the committers SVN repository"	"If your local revision is between my commits an SVN update should do the trick." (Stanbol, 2011-07-08)	0.36
Collaboration/ Help	Help, got, doing	Acts of assistance and volunteering that are central to OSS collaboration	"IPMC members are free to volunteer to mentor a project, to do so they should mail the project stating their intentions."	"Maybe I can help with that." (OpenOffice, 2012-05-29)	0.07
Artifact Management*	Artifact, Artifacts, Nexus[7]	Activities around development, maintenance, and deployment of artifacts	"Release artifacts must include 'incubating' in the final file name"	"As [NAME] said next step is to build and post artifacts" (Etch, 2010-09-27)	0.35
Apache Meetups/ Conferences	Meetings, Meetup, Conference	Conferences, meetups as well as official meetings among ASFI developers and organizers	"For inclusion in the agenda for the next board[5] meeting a resolution should be submitted at least 72 hours before that meeting"	"The board meeting scheduled for Wed 15 July 2015, 10 : 30 AM Pacific" (Sentry, 2015-07-02)	0.32

[1]ASFI projects generally graduate into an independent TLP (top-level project) or as a subject of an existing project
[2]Whimsy is an Apache utility used for managing mailing lists
[3]The form comprised an email header format that could not be compared to email content. We retained this portion of the policy, which provides sufficient context to detect discussions around the use of appropriate paperwork in PPMC elections
[4]Month-year page refers to an indexing system used by the ASFI to collect and track project reports as they become due.
[5]The Apache Software Foundation board
[6]Apache Dubbo was one of the largest projects in our dataset, with frequent activity in this topic
[7]Nexus is a repository management tool