DEPARTMENT OF APPLIED IT

FACULTY OF SCIENCE AND TECHNOLOGY

Negotiating AI in Open Source Software Communities

A Case Study of the LLVM Project

Maria Savva

ESSAY/THESIS:	**30 HP**
Programme:	MSc in Communication
Level:	Second Cycle
Term/year:	Spring Term 2026
Supervisor:	Alan Said
Examiners:	Charlott Sellberg
Word count:	10166

Abstract

Open source software (OSS) communities sustain critical infrastructure through volunteer contribution and review, yet AI coding tools are placing new pressure on the social mechanisms that reproduce these practices. This study examines how one such community collectively deliberates on the creation of an AI tool policy, examining how apprenticeship-based learning communities adapt to AI technologies. Prior research on AI in OSS has concentrated on technical and legal issues, with less attention given to how developers themselves articulate and respond to AI's integration. This study draws on Communities of Practice and Legitimate Peripheral Participation, frameworks to explore how situated learning can continue when newcomers depend their learning on AI tools rather than engaging directly with the community's core members.

Employing a qualitative case study design, this research conducted a thematic analysis of 195 replies across four Request for Comments (RFC) threads from the LLVM Discourse forum, posted between May 2024 and February 2026. Participants were classified into groups based on forum trust levels and self-referential indicators within their replies. The findings reveal that four concerns persisted across the deliberation*, Code Understanding*, *Accountability*, *Reviewer Protection*, and *AI Use Disclosure*. Positioning varied by participation level, with Leaders acting as moderators, Core Members defending review as a relational mechanism, and Peripheral Participants negotiating their entry on contested terms. The tension of AI integration falls at the core-peripheral interface, where situated learning takes place. The principle of “human in the loop” is emerging as a shared repertoire articulating what these communities need to remain reproducible under AI pressure.

Keywords: open-source software, LLVM, Communities of Practice, AI tool policy, thematic analysis

Acknowledgements

To my supervisor, Alan Said, thank you for asking the questions and leaving me to find the answers. That kind of supervision teaches more than instruction ever could.

To my husband, one of the most rigorous thinkers I know and, more importantly, my best friend, thank you for making the long stretches of writing less lonely. You were the steady presence in the background, and that meant more than these few lines can say.

And to my daughter, still in the belly, thank you for being patient with me while I finished this thesis. You waited quietly while I worked, giving me kicks of confirmation at every step,and I cannot wait to meet you.

List of Abbreviations

AI	Artificial Intelligence
CoP	Communities of Practice
LLM	Large Language Model
LPP	Legitimate Peripheral Participation
OSS	Open-Source Software
PR	Pull Request
RFC	Request for Comments
VCoP	Virtual Communities of Practice

Table of Contents

List of Tables

List of Figures

Introduction

Open-source software (OSS) projects form a critical part of the global technology infrastructure, sustained by communities of volunteer contributors who develop and maintain the codebase and review submissions (Fang & Neufeld, 2009; Lakhani & von Hippel, 2003). These community members hold different roles and levels of responsibility, ranging from peripheral contributors to core members, maintainers, who are responsible for reviewing all new code submitted to the project and ensuring its overall quality (Ye & Kishida, 2003). This review capacity is finite, and recent shifts in how code is produced are placing new strain on it. As AI tools increase the volume of code submissions, concerns have emerged regarding the quality and reliability of such contributions (Xu et al., 2025). This increase places an additional burden on those core members, who must review growing amounts of code. Left unaddressed, these pressures risk disrupting the long-term resilience of the OSS ecosystem (Feng et al., 2026).

These are not merely technical issues but questions that communities collectively negotiate and try to resolve. AI tools have offered several benefits for software development, but their integration into OSS projects has simultaneously raised critical concerns around ethical use, transparency, and trust (Ahuja et al., 2025). A growing body of research has focused on the technical challenges that AI code generation introduces (Masaya Osaki, 2008; Mbizo et al., 2024; Stalnaker et al., 2026), yet limited attention has been given to how OSS communities collectively discuss and respond to the challenges that AI brings to their projects. As projects begin developing AI-specific policies and guidelines, understanding how these communities negotiate such decisions becomes an important area of inquiry.

A particularly rich example of such deliberation is taking place within the LLVM project, a widely used open-source compiler infrastructure that translates programming language to machine language (Li & Jiao, 2023). What began as a research initiative at the University of Illinois has grown into an umbrella project hosting multiple subprojects, with wide application across commercial, open-source, and academic contexts (The LLVM Compiler Infrastructure, n.d.-b), and the support of major technology companies including Apple, Google, and NVIDIA (LLVM Foundation, n.d.). The project has a dedicated forum, where community members discuss a variety of issues regarding the usage and development of the project (LLVM Discussion Forums, n.d.-a). In 2024, the community deliberated a proposal to formalize its governance through elected bodies and a project council acting as a final decision-making authority, reflecting an emphasis on transparent, community-driven decision-making (beanz, 2024).

The project maintains a dedicated discussion forum that hosts long-term asynchronous discussions - commonly referred to as threads (LLVM Discussion Forums, n.d.-b). While OSS projects employ various governance mechanisms, such as the voting systems used by the Apache Software Foundation (Wang et al., 2023), LLVM relies on open community deliberation through its Request for Comments (RFC) process (The LLVM Compiler Infrastructure, n.d.-d), where a proposal is shared with the community for open discussion before a decision is reached, and the discussion itself shapes how the proposal evolves. RFCs are visible to all members and the public. Over the past two years, an extensive series of RFC threads has been dedicated to the creation of an AI tool policy, making visible how an OSS community collectively works through a contested issue and shapes policy through deliberation.

To examine this deliberation, this study draws on the theoretical framework of Communities of Practice (CoP) (Lave & Wenger, 1991) and adopts a qualitative approach to examine the concerns and perspectives that emerge among participants in the LLVM discourse forum, as they deliberate on the AI tool policy through the RFC process. In doing so, it explores how AI tools may reshape the situated learning and participation that sustain these communities. Because members with different roles and levels of participation shape this policy, the study also examines whether position within the community influences the stance a member takes.

The study proceeds with a literature review covering CoP and LPP as the theoretical grounding alongside relevant literature on AI in software development and OSS communities, followed by a methods chapter detailing the qualitative approach, data collection, and ethical considerations. A data analysis chapter then describes the corpus, the participant classification, and the analytical steps. The findings are presented in two parts, addressing each research question in turn. A discussion situates the results in relation to the existing literature, and the thesis closes with a conclusion outlining the study's contribution, limitations and future research.

Literature Review

To examine how the LLVM community deliberates on the AI tool policy, it is first necessary to establish some foundational concepts. This chapter introduces the theoretical framework of CoP and situated learning, defines what is considered as online communities and OSS communities, outlines the governance models that shape decision-making in OSS communities and presents how AI has shaped software development and OSS communities today. The chapter closes by presenting the research gap this study aims to fill and the research questions.

Communities of Practice

CoP is a phenomenon that gained formal recognition when introduced by Lave and Wenger (1991) in their work on situated learning. The term describes groups of people who share a concern or passion for something and engage in a process of collective learning through regular interaction (Wenger-Trayner & Wenger-Trayner, 2015). Three elements define a CoP and distinguish it from other types of communities. The first is the domain, a shared area of interest or expertise that gives the community its identity, such as software development or nursing. The second is the community, the social relationships and interactions through which members engage with one another around their shared domain. The third is the practice, the shared repertoire of resources, methods, and knowledge that members develop over time as active practitioners (Wenger-Trayner & Wenger-Trayner, 2015). Through sustained interaction, this repertoire, including common language, tools, artifacts, and ways of doing things, is continuously negotiated as the community evolves (Wenger, 1998).

Wenger (1998) further elaborates how the community and practice elements cohere through three interrelated dimensions: mutual engagement, the relationships and ongoing interactions through which members do things together and through which the community is constituted as a community rather than as a collection of individuals sharing a domain; joint enterprise, the negotiated shared understanding of what the community is about, continuously renegotiated as conditions change; and the shared repertoire introduced above. Of these, mutual engagement and shared repertoire are the dimensions on which this study draws to interpret the LLVM deliberation: shared repertoire to read what the community articulates as the conditions for its sustained practice, and mutual engagement to read how members positioned themselves across levels of participation.

Lastly, CoPs vary widely in size, formality, and medium of interaction, ranging from small informal groups to large, distributed communities operating online (Wenger-Trayner & Wenger-Trayner, 2015). Members do not all participate equally, a dynamic explored in the following section on situated learning.

Situated Learning: Legitimate Peripheral Participation

Situated learning theory, as developed by Lave and Wenger (1991), proposes that learning is not an individual cognitive process but an “integral and inseparable aspect of social practice” (p. 31). Drawing on the concept of apprenticeship, Lave and Wenger (1991) introduced the notion of Legitimate Peripheral Participation (LPP) to describe how newcomers gradually become full members of a community. The learning process in a CoP is not structured through formal instruction. Instead, newcomers learn by participating in the everyday activities of the community and gradually taking on more complex tasks. Through this involvement, they develop relationships with more experienced members, who model

practices and provide feedback. Importantly, no single individual holds authority; rather, it is distributed based on how the community itself is organized. For this transition from periphery to full participation, newcomers need access to the ongoing activities of the community through opportunities for participation (Lave & Wenger, 1991). While this access is vital for the sustainability of the community, it can also be challenging because it requires active engagement from the newcomer and depends on the community's willingness to create those opportunities.

Situated learning in OSS communities

OSS communities offer a clear illustration of this process (Oliveira et al., 2025). Newcomers typically begin with small contributions such as reporting bugs or submitting minor fixes, receive feedback from experienced reviewers, and over time take on more significant responsibilities within the project. The governance model of the project shapes how this process unfolds, defining who reviews contributions, how feedback is given, and what pathways exist for newcomers to advance. In this sense, the organization of the community, rather than any individual maintainer, structures the learning process.

The introduction of AI tools into this process raises questions about the conditions under which situated learning occurs. When newcomers use AI to generate contributions, they may skip the hands-on experience that traditionally drives learning within the community. This raises a fundamental question: can situated learning still occur when newcomers generate contributions without directly engaging with the community's core practices?

Online Communities

As digital technologies enabled remote collaboration, CoP moved to virtual environments, often referred to as Virtual Communities of Practice (VCoP) (Nistor et al., 2014). The core principles of VCoP remain the same: domain, community, and practice, but here interaction occurs through digital platforms. However, not every group interacting online constitutes a community. Herring (2004) proposed a set of criteria for identifying genuine online communities, distinguishing them from loose collections of individuals on a platform. According to these criteria, an online community is characterized by active and self-sustaining participation; the emergence of roles; rituals, and hierarchies; evidence of shared history; culture, norms, and values; collective self-awareness as a distinct group; expressions of solidarity and support; and the presence of criticism, conflict, and means of conflict resolution.

By Herring's (2004) criteria, LLVM qualifies as a genuine online community: it has formalized conflict resolution through the RFC process, role hierarchies (Ye & Kishida, 2003), collective self-awareness as “the LLVM community”, and shared norms codified in its Code of Conduct (The LLVM Compiler Infrastructure, n.d.-a). Combined with its shared domain and collectively developed practices, this makes it a Virtual Community of Practice.

OSS communities

As briefly mentioned in the introduction, OSS communities are sustained by volunteers who take on different roles, ranging from developers who contribute code to maintainers who oversee the project's direction and quality. This volunteer base constitutes the core of OSS projects (Singh et al., 2024). Because the codebase is publicly accessible, contributors from diverse backgrounds can participate and drive innovation. However, this openness also

introduces the risk of low-quality submissions if “unrestricted” (Tan et al., 2024). OSS communities manage this risk by extending trust to contributors gradually, as they demonstrate competence and engagement within the project.

This trust is represented by commit rights, the permission to directly modify the project's main codebase, which is granted to contributors who demonstrate both technical competence and consistent social engagement. Among those with commit rights, some take on the role of maintainer, carrying broader responsibility for the project's direction. This core group evolves over time, allowing new members to bring fresh perspectives and innovation. However, attaining this level of trust is therefore not purely a technical achievement but a social one as well (Tan et al., 2024).

Those who do earn this trust and take on the role of maintainer carry responsibilities that go well beyond technical tasks. Although maintainers are usually experienced developers, what distinguishes an effective maintainer is not technical skill alone but rather the ability to communicate, build community, and guide the project's direction over time (Dias et al., 2021). In practice, maintainers serve a dual function: they ensure the quality of contributions entering the project while also creating the conditions for others to participate and grow within the community.

This participation structure closely reflects the dynamics described in CoP theory. The shared domain is the software project itself; the community is formed through the social relationships among contributors; and the practice consists of the development activities, review processes, and governance mechanisms through which the project evolves. The progression from newcomer to committer to maintainer parallels LPP, where newcomers gradually move toward full membership through sustained engagement with the

community's practices. The way participation unfolds is closely linked to how projects structure authority and decision-making, which will be discussed in the following section.

OSS Governance

Governance structures define how authority, responsibility, and decision-making processes are distributed within the community (Oliveira et al., 2025). Governance models range from benevolent dictatorship and meritocracy to foundation-backed democratic structures, and projects further differ in how they are supported, operating within a company, seeking external funding or being maintained by individuals (Eghbal, 2020).

These arrangements shape pathways of participation, whose voices carry weight, and how members advance, and well-established frameworks are linked to higher contributor participation and more sustainable community dynamics (Oliveira et al., 2025). Some projects rely on open deliberation, where proposals are shared publicly, and members negotiate collectively. Understanding how such deliberation functions is particularly relevant as OSS communities face the broader challenge of integrating generative AI into their established practices.

AI and software development

The impact of AI tools on code quality remains contested. It is argued that code produced through human-AI collaboration lasts longer than code from fully autonomous agents, though this does not necessarily indicate higher quality, since developers may avoid editing code they did not write, or defects may not yet have been discovered (Rahman & Shihab, 2026). Additionally, practitioners report other practical concerns, such reliability issues, misleading outputs, and security vulnerabilities, acknowledging that current AI tools cannot be trusted without careful human oversight (Mbizo et al., 2024). The value of AI-

generated code therefore depends heavily on how the tools are used and the level of human involvement.

In addition to these technical concerns, a gap emerges between the pace of AI adoption and the organizational preparedness to manage it. Although the majority of developers have integrated AI tools into their everyday workflows, many encounter company restrictions driven largely by fear of information leakage (Stalnaker et al., 2026). More critically, a significant number of practitioners report that their organizations provide no documentation or guidelines around the use of generative AI, leaving questions of responsible use, copyright compliance, and code ownership unaddressed (Stalnaker et al., 2026). This lack of organizational guidance points to a broader challenge: as AI tools become embedded in software development practices, the need for clear policies and AI literacy becomes increasingly urgent, not only within industry but across other domains as well. In computing education, for instance, the need for policy frameworks has become vital, aiming to ensure that future practitioners engage with AI tools in ethically and legally responsible ways (Bailey, 2024).

AI in OSS communities

This need for clear policies and organizational preparedness is particularly pressing in open-source software communities, where the impact of AI starts to become more and more evident. As AI automates tasks that traditionally served as opportunities for collaboration and learning, such as code review and mentorship, the social interactions through which community members develop shared understanding and newcomers gain experience risk being diminished. Replacing human-centered workflows in OSS communities may fundamentally reshape social dynamics, community norms, and participation structures

(Feng et al., 2026). Furthermore, since the launch of GitHub Copilot in 2022, a notable rise in code contributions has been observed, accompanied by a significant increase in maintenance-related workload for core developers. Reviewing, revising, and often discarding submitted code contribute to growing technical debt in OSS projects, the accumulation of unreviewed or low-quality code that requires future effort to fix (Xu et al., 2025).

However, research also identifies benefits. AI tools support pull request writing and review, reducing review time and increasing the likelihood of acceptance (Xiao et al., 2024), and have been proposed as a resource for newcomer onboarding (Santos et al., 2025). Tan et al. (2025) similarly argue for AI-driven mentorship. Such tools may also reshape the newcomer-expert relationship through which knowledge is transferred, a concern that connects directly to LPP: AI may alter how newcomers learn, gain access, and become full participants.

Research Questions

While the studies reviewed above identify various technical, social, and ethical concerns around AI in software development and OSS communities, research examining how OSS communities develop AI-specific policies remains scarce. This study addresses that gap by examining how the LLVM community deliberates AI tool policy through its RFC process, with attention to both the concerns and perspectives that emerge in these discussions and the positions of members with different levels of participation. Specifically, this study aims to answer the following questions:

RQ1: What concerns and perspectives emerge among LLVM community members deliberating around the creation of an AI tool policy?

RQ2: How do community members with different levels of participation position themselves in these deliberations?

The following chapter details the research approach adopted to address these questions.

Methods

This chapter presents the research approach and methodology used to examine how an OSS community deliberates around its AI tool policy. Firstly it introduces and justifies the qualitative approach adopted for the study and the analytical method through which data were examined. Then describes the case selection and data collection process, and lastly the ethical considerations of the study.

A Qualitative, Theory-Informed Approach

This study follows a qualitative approach, as the research questions require an inductive, interpretive understanding of how discussants express, negotiate, and contest their perspectives on the AI policy. To answer those questions, a single case study design was adopted, focusing on the LLVM discourse forum deliberation around its AI tool policy as a bounded, real-world instance of this phenomenon. Qualitative methods are appropriate when the aim is to explore meaning and process in depth, which cannot be captured solely through quantitative methods (Clark et al., 2021).

Qualitative analysis is fundamentally based on categorization, where observations become patterns to be analyzed (Treadwell & Davis, 2020). Of the three coding approaches, fixed, flexible, and grounded-in-data, this study adopted flexible coding. The study is grounded in a constructionist perspective, which understands social phenomena as constructed through interactions and meaning-making within communities (Clark et al., 2021). The interpretive nature of the analytical process is acknowledged as a limitation of the approach.

Within this approach, thematic analysis was selected as the analytical method of the study. Unlike more structured approaches such as grounded theory or critical discourse

analysis, thematic analysis does not follow a fixed set of techniques, making it adaptable to the specific needs of the research (Braun & Clarke, 2006). This flexibility is particularly suited to analyzing exploratory data such as online forum discussions, where the boundaries of relevant themes cannot be fully anticipated in advance.

This study employs thematic analysis at a theoretical and latent level (Braun & Clarke, 2006). Rather than purely inductive coding, it was informed by CoP and LPP while remaining responsive to themes emerging from the data. Consistent with a constructionist perspective, the analysis moved beyond the surface content to examine the underlying positions through which members negotiated the AI policy. This grounding aims to address a known limitation of thematic analysis without an anchoring framework, that analysis risks remaining purely descriptive (Braun & Clarke, 2006).

Case Selection and Data Collection

The LLVM project was selected as the case for this study based on several criteria. It is a widely used open-source project with notable industry involvement that maintains a publicly accessible discourse forum and has undergone an extensive deliberation process around AI policy. While other OSS communities have also engaged with questions around AI policies, LLVM offers a particularly rich case given the volume, depth, and transparency of its public discussions on the topic. The selection follows a convenience and purposive sampling approach, based on the case's accessibility and relevance to the research questions (Clark et al., 2021). The dataset consists of publicly available RFC threads on the project's AI tool policy, were collected in February 2026.

Ethical Considerations

The data consists of publicly available forum posts from the LLVM discussion forum. The project's Developer Policy explicitly states that the discourse forums are public and archived, and that notices of confidentiality or non-disclosure cannot be respected (The LLVM Compiler Infrastructure, n.d.-c). As such, contributors participate with the awareness that their posts are publicly accessible.

Despite the public nature of the data, measures were taken to protect the identities of the discussants. They were assigned anonymized labels based on their participation level rather than being identified by their usernames. It is nonetheless acknowledged that verbatim quotations, which are used in the findings, carry a re-identification risk, as quoted posts can potentially be traced back to their authors (Bouvier, 2022), a common concern when working with public online data.

The analytical focus remained on patterns of deliberation rather than profiling individual contributors, and no part of the analysis was intended to expose or harm any community member (Bouvier, 2022). The study involved no direct interaction with participants and no access to private content.

Data Analysis

This chapter presents the analytical work conducted on the data prior to reporting the findings. Firstly, the chapter describes the corpus and how it was organized for analysis, then it explains how participants were classified into participation groups and lastly, it details how themes and participants' positioning were analyzed.

The RFC Thread Corpus

The corpus consists of four RFC threads from the LLVM discourse forum: “RFC: Define policy on AI tool usage in contributions” (RFC 1) (rnk, 2024), “Our AI policy vs code of conduct and vs reality” (RFC 2) (JonChesterfield, 2025), “[RFC] LLVM AI tool policy: start small, no slop” (RFC 3) (rnk, 2025b), and “[RFC] LLVM AI tool policy: human in the loop” (RFC 4) (rnk, 2025a).

The four threads contain 195 replies spanning from May 2024 to February 2026, covering the deliberation process from the initial proposal through successive rounds of revision. The corpus includes both conversational replies and deliberative artefacts embedded within them, such as the draft policy texts found in the initial comments of some RFCs.

Data Organization

The dataset was organized to support both stages of the analysis. Each RFC thread was kept separately, with its replies recorded in the order they appeared. For each reply, basic information was noted, the post number, date, contributor's username, and forum trust level, alongside the codes, themes, and interpretive notes generated during the

analysis. Additional analytical layers were added during the analysis itself, as described in the following section.

Participant Classification

To support the analysis of "RQ2: *How do community members with different levels of participation position themselves in these deliberations?*", each discussant was classified into a participation group as a preliminary analytical step. The classification was based primarily on the trust level designation visible on each member's public forum profile. The LLVM Discourse forum assigns members one of four trust levels: Basic, Member, Regular, and Leader. Participants with hidden profiles could not be classified and were therefore excluded from the analysis.

Differentiating the Member Trust Level

Upon reading the discussion replies, discussants at the Member trust level appeared to be considerably more diverse than the others, with comments revealing variation in role and expertise. To capture this variation, the Member trust levels was further differentiated into four sub-types, Reviewer, Experienced, Newcomer, and Unclassified, based on language indicators and self-referential statements within the replies, such as references to one's role in the project or level of technical expertise. The Unclassified sub-type covers discussants, for whom no further differentiation could be determined from the replies. This classification process inevitably involved some degree of interpretive judgment in ambiguous cases, which is acknowledged as a limitation of the study.

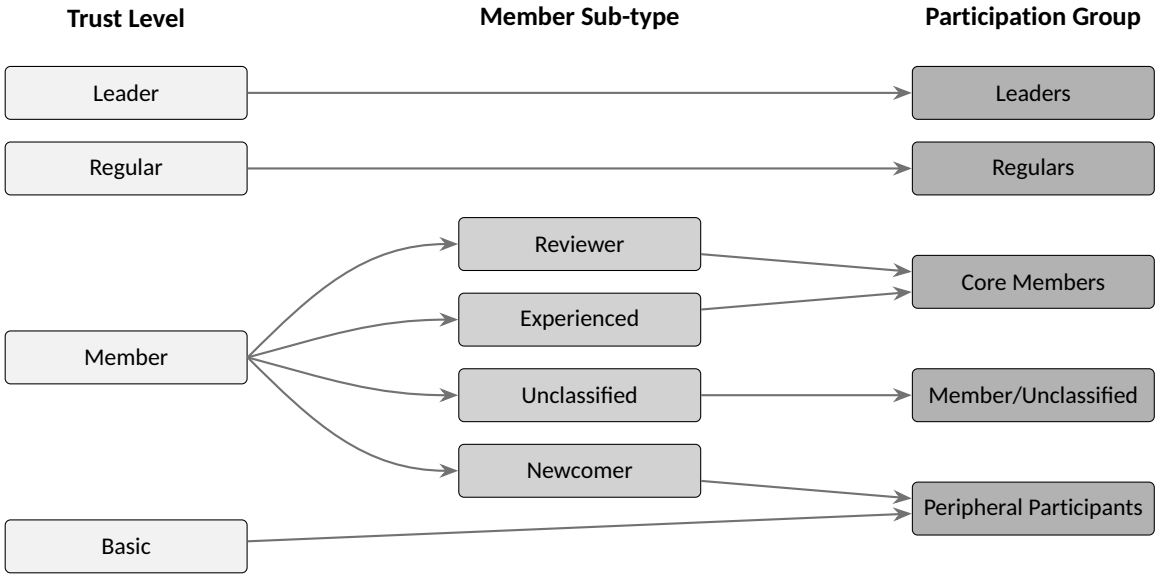
Consolidation into Participation Groups

Figure 1 illustrates how the four trust levels assigned by the LLVM Discourse forum were transformed into the five participation groups used in the RQ2 analysis through three

steps. First, the Leader and Regular trust levels mapped directly onto the Leaders and Regulars groups, since members at these levels were sufficiently distinct on their own. Second, the Member trust level was differentiated into four sub-types (Reviewer, Experienced, Unclassified, Newcomer), as described in 4.3.1. Third, these sub-types were consolidated with the Basic trust level into three further groups. Reviewer and Experienced sub-types were combined into Core Members. The Unclassified sub-type was retained as its own group, labelled Member/Unclassified. The Newcomer sub-type was combined with the Basic trust level into Peripheral Participants. Each member was assigned an anonymized label reflecting their participation group to protect their identity.

Figure 1

Categorical Flow Diagram of the participant classification process



Analyzing Themes and Positioning

For addressing RQ1, identifying the concerns and perspectives emerging from the deliberation around the AI tool policy", the analysis began with repeated reading of all four RFC threads to develop familiarity with the data. Following Braun & Clarke 's (2006) phases

of thematic analysis, initial codes were then generated inductively, labeling the concerns and perspectives expressed by community members, operating primarily at a semantic level. These codes were then grouped into broader themes, reviewed and refined iteratively to ensure they accurately represented the patterns present across the dataset, then organized into a consistent coding scheme and applied across all four RFC threads.

To focus the analysis on themes with sufficient presence in the data, a frequency threshold of n>3 was applied per RFC thread. One theme, *Technical*, was excluded from the qualitative analysis despite meeting the threshold, as its content related to implementation details rather than deliberation. The remaining themes were examined chronologically per RFC thread, tracing how the deliberation evolved from the initial phase to the resolution of the policy.

For RQ2, examining how community members with different levels of participation position themselves in these deliberations, the analysis shifted to a more interpretive, latent level (Braun & Clarke, 2006), informed by CoP as a theoretical grounding. The dataset was re-examined by reading each comment across all four RFC threads in sequence, tracking how every contributor positioned themselves throughout. Each contribution was coded with a short label capturing the stance being expressed and then these labels were organized per participation group. Whether members of a group would converge on a shared stance or express a range of positions was treated as an open question, answered by the data itself; representative quotes were then selected to illustrate the patterns within each group.

Findings

This chapter presents the results of the thematic analysis of LLVM RFC discussion threads. The analysis produced 39 unique themes across the corpus. The findings are organized around the two research questions. This chapter reports the findings for RQ1 (concerns and perspectives across the deliberation), tracing how themes emerged and evolved across the four RFC threads chronologically, and follows with the findings for RQ2 (positioning across participation levels), examining how participants at different levels engaged with the deliberation.

Themes by RFC Thread

This section presents the concerns and perspectives that emerged across the four RFC threads. The subsections that follow describe how each of these themes was discussed in the thread, illustrated with representative quotes drawn from the data.

RFC 1: "Define policy on AI tool usage"

In the opening RFC, participants debated who should be responsible for managing the risks that AI tool use introduces into contributions. *Accountability* and *Reviewer Protection* were the most frequently occurring themes. Those in favor of a permissive policy argued that contributor accountability under existing standards was sufficient, while others called for mandatory disclosure and stronger guidelines, describing maintainer attention as a limited collective resource. *Copyright Issues* also appeared, though less frequently, with participants pointing to unresolved legal uncertainty around AI-generated code. Representative quotes are presented in Table 1.

Table 1

RFC 1: Representative Quotes per Theme

Accountability
"It's important to stress that contributors are responsible for the quality and content of code they submit, AI generated or not."
Reviewer Protection
"The reviewer is put in the awkward position of having to ask 'did you write this yourself?' without a realistic way to verify." "Maintainer attention is our most precious, scarce resource"
Permissive Policy
"I'm essentially proposing a liberal policy..."
AI use disclosure
"I am not asking for a ban; I am asking for disclosure."
Copyright Issues
"There is no settled case law in the United States on how copyrights are handled relating to the output of such programs."

RFC 2: "Our AI policy vs code of conduct and vs reality."

In RFC 2 a wider variety of themes emerged. While the core concerns from RFC 1, *Accountability*, *Reviewer Protection*, and *AI Use Disclosure* persisted, the discussion broadened to include questions about community identity, trust, and the role of newcomers, as reflected in the representative quotes shown in Table 2. A concrete case involving a newcomer who had used AI in their contribution prompted further discussion of these themes, with the participant arguing that the source of the quality problems was inexperience rather than AI use itself. Participants identified three priorities, quality, reviewer time, and inclusion, and noted inclusion as the one most likely to be compromised. *Trust* emerged as a new theme, with participants noting that AI use makes it harder to know whether they are genuinely interacting with another human, and that no policy can substitute for honesty if contributors choose not to disclose AI use. Legal and licensing concerns were also present, particularly around whether LLM training data could introduce licensing conflicts. Cross-project comparisons to Fedora, Curl, and NetBSD featured throughout.

Table 2

RFC 2: Representative Quotes per Theme

Accountability
"The contributor claims ownership of that content, regardless of how it was generated… and as such, the contributor takes responsibility for that content."
Reviewer Protection
"Contributions written without a strong attempt at understanding are extractive. They do not provide sufficient value relative to the amount of time that they take to review."
AI use disclosure
"I also just edited the PR's original body to disclose AI usage." "The lack of required disclosure leaves reviewers wondering if we are interacting with a real human being." "I don't think we should require disclosure where AI has been used to help explore and understand the code…"
Code quality regardless of origin
I'd say the main reason the code is bad quality is because I wrote it less than 3 months after graduating" "We should focus on the quality of contributions and not the tool used to generate them."
Code Understanding
"If someone has taken time to study code, stands by it as if it were their own, and will personally engage in the back and forth review process then I don't mind if they wrote it from scratch.
Community Inclusion & Participation
"How we can 1) maintain quality standards, 2) without undue burden on reviewers, 3) while being welcoming to newcomers…3) is probably the point we're most willing to compromise on, by spending less effort trying to salvage AI slop from new contributors."
Trust
"I would prefer to not end up with a policy that may reward 'dishonesty'." "If people are lying about everything… there aren't really any rules that are going to help us there anyway."
Newcomers as primary AI tool users
"Either to reject newcomers or to reject AI contributions." "Do we ban interns too so they never get any experience due to quality concerns?" "AI is a means to bridge the knowledge gap for me."
Anti AI
"I would prefer to see LLVM adopt a no-AI policy on quality grounds."

RFC 3: "LLVM AI tool policy: start small, no slop"

In RFC 3, the leader put forward a draft policy, and participants engaged directly with the proposed text. *Reviewer Protection* and *Code Understanding* remained the most frequently occurring themes, with discussion now centering on the specific provisions of the draft. *Accountability* appeared again, with participants stating that contributors should be able to explain and defend their submissions*. Policy Endorsement* appeared for the first time above threshold, with several participants expressing acceptance of the draft, often with reservations. The most contested element was the 'extractive' label: some participants accepted it as an enforcement tool, while others questioned whether it was a fair governance mechanism, particularly for contributors who lacked the expertise to know in

advance what would be considered unacceptable. *Policy Framing and Governance* also appeared, with participants debating how the policy should be designed and applied. The thread ended with the leader announcing a revised draft and new RFC thread. Examples of representative themes are given in Table 3.

Table 3

RFC 3: Representative Quotes per Theme

Reviewer Protection
Currently, LLVM lacks reviewer capacity to deal with regular PRs…" "Transforms that do not have real-world usefulness provide negative value to the LLVM project, by taking up valuable reviewer time."
Code Understanding
"What makes changes valuable long-term is the understanding" "If authors cannot explain how their code works to any reviewers, then what is the hope in having those same authors and reviewers maintaining that code?"
Policy Revision
"I made what I consider substantial changes... so I created a new RFC thread to follow up on this."
Policy Endorsement
"I think your draft nicely encapsulates a balanced set of feedback and expectations for the community."
Accountability
"To submit to the project you need to take full ownership of your submission... If someone is not able to properly describe and motivate their changes, then there's not a path forwards for a successful review anyway."

RFC 4: "LLVM AI tool policy: human in the loop"

RFC 4 was the last thread in the deliberation. The draft policy text, pasted in full within the opening reply, made "human in the loop" its central concept, including the statement that "passing maintainer feedback to an LLM doesn't help anyone grow, and does not sustain our community."

Policy Revision has been the most frequent theme, with the leader presenting an updated draft and participants continuing to propose adjustments, though the overall direction was no longer in dispute. *Code Understanding* was reaffirmed in relation to both the draft's provisions and broader community needs. A dissenting voice challenged the ownership requirement, arguing that demanding full accountability before participation

excludes contributors who could develop ownership through engagement. A policy violation incident also featured in the discussion: a merged PR with undisclosed AI use prompted a public dispute with participants disagreeing on whether a violation had occurred and noting that the policy's language around disclosure was insufficiently clear. *Reviewer Protection* appeared again, particularly in concerns about the time and effort required to review AI-assisted PRs. Participants also referenced cross-project comparisons to IREE, and ASF communities. The thread ended with the formal acceptance of the policy by the Project Council. Representative quotes are presented in Table 4.

Table 4

RFC 4: Representative Quotes per Theme

Policy Revision
"I got a lot of feedback from various meetings on the proposed LLVM AI contribution policy, and I made some significant changes based on that feedback."
AI use disclosure
"Can we have a standardized policy of asking all new contributors to declare in their pull request that they are compliant with the AI policy."
Code Understanding
"Requiring a human in the loop who understands their contribution well enough to answer questions about it during review"
Policy Violation
"There seems to be a case of our own maintainers not following the policy that we have adopted." "There is no violation of the AI policy or its intended goal."
Accountability
"The contributor is always the author and is fully accountable for their contributions." "Ownership is not a ticket that one must purchase before entering the park. Ownership is the outcome of a successful collaborative journey."
Policy Endorsement
"This RFC was discussed at the Project Council meeting. We found broad community support for the proposal... The proposal is now accepted."
Reviewer Protection
"Determining whether a PR actually meets those standards still costs maintainer time and attention, and I strongly suspect that cost is often close to what it would take to just do a full review anyway."
Cross-project comparison
"Worth noting that LLama now has a human in the loop policy as well." "We adopted the human in the loop policy in IREE too."

Theme distribution across RFCs

Across all four RFC threads, as it appears in Table 5, three themes appeared above the frequency threshold: *Accountability, Reviewer Protection*, and *Code Understanding*. Other themes appeared in specific threads only. *Community Inclusion & Participation*, *Trust*, and *Newcomers as Primary AI Tool Users* appeared in RFC 2, in connection with the case of a newcomer's contribution, but did not appear above threshold in the subsequent threads. *Policy Endorsement* appeared from RFC 3 onwards, after a draft policy was first presented. *Policy Violation* appeared in RFC 4 only, in connection with the merged PR incident.

Table 5

Most Frequently Occurring Themes per RFC Discussion Thread (n>3)

Theme	RFC 1	RFC 2	RFC 3	RFC 4	Total
Reviewer Protection	6	7	**9**	4	**26**
Accountability	**8**	**8**	4	6	**26**
Policy Revision	—	5	7	**12**	**24**
Code Understanding	—	6	**8**	7	**21**
AI Use Disclosure	5	7	—	**8**	**20**
Policy Endorsement	—	—	5	6	**11**
Cross-project Comparison	—	6	—	4	**10**
Policy Framing & Governance	—	4	4	—	**8**
Policy Violation	—	—	—	7	7
Code Quality Regardless of Origin	—	7	—	—	7
Community Inclusion	—	5	—	—	5
Trust	—	5	—	—	5
Permissive Policy	5	—	—	—	5
Technical	—	—	—	5	5
Legal & Licensing Concerns	—	4	—	—	4
Newcomers as Primary AI Users	—	4	—	—	4
Anti AI	—	4	—	—	4
Copyright Issues	4	—	—	—	4

Note. "—" indicates the theme appeared <3 times in that RFC or not at all.

Positioning Across Participation Levels

This section presents how participants at different levels of participation positioned themselves across the four RFC threads. The subsections that follow are organized by participation group: Leaders, Regulars, Core Members, Peripheral Participants, and Member/Unclassified. For each group, key positions are illustrated with representative quotes drawn from the data. Further representative quotes for each group are presented in Appendix.

Leaders

Across the four threads, L1 summarized feedback, proposed institutional mechanisms such as a roundtable, and refined the policy text based on community input. L1 started in RFC 1 by arguing for broad permissibility of AI tool use, and by RFC 4 aligned with the "human in the loop" principle, framing the draft as enabling contributors using "LLM assistance to increase their productivity" while giving "maintainers a useful policy tool for pushing back against unwanted contributions."

Other leaders took different positions. L4 distinguished acceptable from unacceptable AI use in RFC 1 and called for a blanket ban by RFC 3, arguing that AI had already been "extractive in terms of limited reviewer and triage resources." L5 challenged L4 directly on copyright responsibility. L6 called for a stricter policy in RFC 3 and argued that experience contributors and newcomers should be subject to different rules. By RFC 4, most leaders aligned behind the final proposal and L11 formally announced that the Project Council had found "broad community support" and that "the proposal is now accepted." (see Appendix, Table A1).

Regulars

Regulars contributed across the threads without taking restrictive or permissive positions on AI tool use itself. R1 identified newcomers as the group most likely to rely heavily on AI tools and proposed that the policy aim at them specifically. R2 argued that the tool is irrelevant if the contributor stands by the code, putting it bluntly that they would not mind if a contributor "wrote it from scratch, used sed, or had a room of monkeys with typewriters write a billion drafts until one worked" provided they engage in review. R3 expressed empathy toward a non-expert contributor (MN2) in RFC 3, and in RFC 4 named the core problem to be "that reviewers struggle with a wave of LLM output coming as contributions, where contributors don't have enough understanding of their PRs (see Appendix, Table A2).

Core Members

Core Members showed considerable internal diversity, ranging from permissive to strongly restrictive positions. MR6 argued across RFCs 2 and 3 that quality rather than tool origin should be the criterion for acceptance. MR9, MR10, ME1, ME2, and ME3 took restrictive or anti-AI positions: MR10 described contributions without genuine understanding as "extractive", and ME2 in RFC 3 strongly opposed "any policy that frames machine-generated contributions as encouraged practices". MR3 argued that mandatory disclosure is a matter of respect for reviewers, while MR14 raised the practical awkwardness of asking contributors about AI use. ME4 argued in RFC 4 that contributions should be considered in terms of how trust and merit are earned publicly in open-source communities, not quality alone. By RFC 4, MR1 pointed to the subjectivity of quality judgments (see Appendix, Table A4).

Peripheral Participants

Peripheral participants contributed across the four RFCs. MN2 was the only contributor across the entire dataset to explicitly position themselves as a non-programmer who uses AI to contribute despite lacking programming experience, asking directly whether contributors like them represent a burden or a potential value to the project. Across RFC 3 and RFC 4, MN2 moved from raising inclusion concerns to proposing measurable and objective quality criteria as alternatives to the subjective judgments they identified as risks of exclusion. MN1 reported negative personal experiences with AI tools and identified newcomers as their primary users.

BN1 was pulled into the deliberation after their own PR became a point of contention in RFC 2, and responded by defending their practice directly, emphasizing the time and care invested in reviewing their AI-assisted submission. Other peripheral participants' positions varied with B1 proposing a disclosure mechanism, B2 raising concerns about accepting AI contributions, and B3 arguing that existing standards were sufficient without additional restrictions (see Appendix, Table A5).

Member/Unclassified

Member/Unclassified contributors took varied positions in the deliberation. MU2 called for empirical evidence about existing AI use among contributors before committing to a policy direction. MU7 drew a comparison with pre-AI contribution practices, suggesting that the standards being discussed were extensions of existing expectations rather than new requirements. MU10 stated that contributors without programming expertise have no place in a project like LLVM. MU11 argued that contributions should be evaluated based on quality and effort rather than tool origin (see Appendix, Table A3).

Discussion

The findings chapter shows a community deliberating on the challenges that AI tools use brings to its practice and norms, while negotiating how to adapt so that its practice remains sustainable. This chapter addresses RQ1 (concerns and perspectives across the deliberation), reading them through CoP theoretical framework, and RQ2 (positioning across participation levels), reading them for what they reveal about LPP in the era of AI tools in software development.

RQ1: Communities of Practice Under AI Pressure

In the literature review chapter, the question is raised of whether situated learning can still occur when newcomers generate contributions without directly engaging with the community's core practices. Read through CoP, the LLVM deliberation can be understood as the community naming the conditions under which it believes situated learning can still take place. The three persistent themes, *Accountability*, *Reviewer Protection*, and *Code Understanding*, articulate practice-level prerequisites, while *AI Use Disclosure* functions as the practical condition that enables meaningful review. The sections that follow develop this reading through the elements of the CoP framework.

Protecting how newcomers become reviewers

In Wenger-Trayner and Wenger-Trayner's (2015) framework, practice is one of the three constitutive elements of a CoP, consisting of a shared repertoire of resources, methods, and knowledge that members develop over time as active practitioners. The three persistent themes in the deliberation map onto this practice element. Those themes highlight what the community sees as essential for the sustainability of the practice.

Firstly, understanding the submitted code becomes a prerequisite for participating in the practice because it allows the contributor to engage meaningfully with reviewer's feedback, and it is through this feedback exchange that they gain experience and move from newcomer to experienced contributor and eventually to reviewer. The importance of understanding the code is present in all RFCs, through comments such as "If authors cannot explain how their code works to any reviewers, then what is the hope in having those same authors and reviewers maintaining that code?" (RFC 3). Then, bound with *Code Understanding* comes *Accountability*, the responsibility contributors should take for their submissions. Taking accountability for their contributions and understanding their code makes the reviewing process meaningful and facilitates a sustainable future for the project's practice through situated learning. As clearly stated by a discussant, "To submit to the project, you need to take full ownership of your submission... If someone is not able to properly describe and motivate their changes, then there's not a path forwards for a successful review anyway." (RFC 3).

Protecting reviewer time, protecting the community

Reviewer Protection emerges as a prominent theme because reviewers are themselves a precondition for situated learning. Without their feedback, newcomers have no path into the practice (Lave & Wenger, 1991). The time experienced members volunteer to read, question, and give feedback is not only a mechanism of quality control but also an act of teaching. As mentioned above, this process is what can grow newcomers into future reviewers. Protecting reviewer time, therefore, protects the mechanism by which the community reproduces its own expert base. It therefore is a matter of survival for the community. The draft policy names this directly, warning that "passing maintainer feedback to an LLM doesn't help anyone grow and does not sustain our community" (RFC 4), while

another participant names maintainer attention as “our most precious, scarce resource” (RFC 1).

Disclosure as the precondition for trust

Lastly, *AI Use Disclosure* functions as a practical condition that enables the other three to operate. AI involvement does not undermine *Code Understanding* or *Accountability*. A contributor who can explain and defend their submission demonstrates both, even if AI produced a first draft. What disclosure adds is the reviewer’s ability to verify rather than assume, preserving the conditions under which trust is earned through technical and social engagement (Tan et al., 2024). This transparency is what builds trust and lets reviewers direct their teaching toward those who can grow from it.

The deliberation surfaced this early, as one participant put it, “I am not asking for a ban, I am asking for disclosure” (RFC 1), framing disclosure not as a restriction but as the minimum basis for a meaningful review. The revised policy in RFC 4 institutionalizes this logic, requiring that “Contributors are expected to be transparent and label contributions that contain substantial amounts of tool-generated content” (RFC 4).

Deliberation as negotiation of shared repertoire

Beyond naming the conditions its practice depends on, the deliberation also does something more. The four RFC threads themselves can also be read as the community articulating and negotiating its shared repertoire. As Wenger (1998) notes, this repertoire is “continuously negotiated as the community evolves”. This process of policy creation evolves this repertoire under the new way of practice that AI tools bring with them. What counts as a legitimate contribution, how reviewers should engage with AI-assisted submissions, what the community owes its newcomers, and what it means to own your code, were all brought

into textual form through the RFC threads and the revised proposal drafts. The deliberation is, in this sense, not separate from the practice but part of how the community sustains that practice under a new pressure.

"Human in the loop" as a shared OSS repertoire

These observations extend the socio-technical concerns raised by Feng et al. (2026) and Xu et al. (2025) by providing an empirical account of how an OSS community articulates and defends the conditions on which its practice depends. Where Feng et al. (2026) warn that replacing human workflows could reshape social dynamics, community norms, and participation structures, the LLVM deliberation shows a community working actively to prevent that reshaping by naming what its situated practice requires to hold humans in the loop. Adding to Xu et al.'s (2025) documentation of the quantitative rise in maintainer workload under AI-assisted contribution, the LLVM deliberation captures how the community is working out how to handle that workload by keeping humans in the loop, acknowledging that reviewer attention is "our most precious, scarce resource" (RFC 1).

Beyond this single community, the cross-project comparisons across RFC 2 and RFC 4, to Fedora, Curl, NetBSD, Llama, IREE, and ASF, suggest that "human in the loop" is emerging as a shared repertoire across OSS CoPs as they negotiate the same practice pressure. This raises the question of how such negotiation unfolds inside a single community, and how do members at different participation levels shape it. The next section takes up this question.

RQ2: Mapping the strain across participation levels

Reading across the positions taken by community members at different participation levels, a structural shift becomes visible. Before AI, the technical difficulty of producing code was itself part of the process through which peripheral members gradually moved toward

fuller participation (Fang & Neufeld, 2009). AI tools lower that barrier, granting newcomers and peripheral participants a new route into contribution while shifting the cost to Core Members.

Reading the participants' positions considering the dimension of mutual engagement (Wenger, 1998) and the LPP theory (Lave & Wenger, 1991), both introduced in the literature review chapter, makes this strain interpretable. Mutual engagement is what holds a CoP together as a community and LPP is the path by which newcomers gradually become full participants through that engagement. The findings chapter shows that AI does not pressure all forms of mutual engagement equally; the tension is observed at the interaction between Core Members and Peripheral Participants, between reviewers and newcomers, where situated learning happens. The next sections make visible how each group expresses this strain.

Leaders: Holding the community together across disagreement

Leaders are positioning themselves as moderators. They are acting at a governance level in the deliberation, holding the community together across positions rather than committing to any single one. From an LPP perspective (Lave & Wenger, 1991), leaders are not simply at the top of a participation hierarchy, but they are the members responsible for sustaining the conditions under which others can move from peripheral to fuller participation.

Their own mutual engagement, enacted at the governance level, is not itself disrupted by AI's integration into the practice. What leaders do from this role is to surface, through the RFC process, the issues members closer to the practice face and how mutual engagement among them is being affected, and to craft a policy that protects it.

Core Members: defending the social mechanism

Core members, contributors closest to the practice, produced the strongest pushback against accepting AI tools in the practice. Their daily exchanges with newcomers and peers in the practice level are mutual engagement that sustains the community. Their reviews, according to LPP, are the mechanism through which newcomers move from periphery to fuller participation.

This is also why their position on AI was rarely neutral. Those who rejected AI tools did so on the grounds that it threatened to overload reviewers and produce contributions whose authors would not be able to mutually engage in the review feedback. Those who accepted AI-assisted contributions did so only under conditions that preserved the relational basis of review.

In both stances, what was being defended was not the production of code but the social mechanism through which the practice reproduces itself, the review, feedback, and teaching that turn newcomers into experienced contributors. Core members' position makes a single interpretive point unmistakable: AI's challenge is felt sharpest at the level of mutual engagement, and it is felt sharpest by those whose participation is most engagement-intensive.

Peripheral Participants: entry on contested terms

Peripheral participants spoke from a personal stake. Their concerns were about whether their movement from periphery to fuller participation could begin at all or would be blocked, and they absorbed the heaviest pressure under this question. Two cases, MN2 and BN1, carry the analytical weight here; the remaining peripheral voices occupy positions already visible elsewhere in the deliberation.

MN2's positioning challenges a basic premise of LPP, which assumes that newcomers move toward fuller participation by engaging in the practice of a community whose domain they already share (Lave & Wenger, 1991). They identified themselves as one of the “PC enthusiast users like me that are willing to contribute time and effort to the best of their abilities but lack the programming experience and intimate knowledge of the LLVM code base” (RFC 2), and named their fear of exclusion in the policy's own terms: 'Are people like me more of a burden or could they bring value to the project? The bar of “generally understand the code they are submitting” is understandable, but that would exclude people like me' (RFC 2). MN2's positioning is significant because it makes explicit a claim usually left implicit in deliberations about contribution policy, that for some members, AI is not a productivity tool but a substitute for the domain competence that LPP takes as the starting point of the trajectory toward fuller participation. Though this view can be aligned with research proposing the use of AI tools for easing newcomers’ onboarding (Santos et al., 2025; Tan et al., 2025), it also creates a gap in the traditional workflow of a CoP and the situated learning process.

BN1's case sits differently and when their PR was cited in RFC 2 as evidence that the combination of newcomer status and AI use was overwhelming reviewers, BN1 defended their place by enacting exactly what the deliberation came to identify as the conditions for legitimate AI-assisted contribution. To the disclosure question they responded directly, “I also just edited the PR's original body to disclose AI usage” (RFC 2). To questions of accountability and review effort, they wrote: “I'm not writing a PR with AI and chucking it over the fence. I spent days reviewing and planning the code” (RFC 2). And they relocated the cause of the submission's weaknesses within the conventional grammar of peripheral participation: “the main reason the code is bad quality is because I wrote it less than 3

months after graduating and with almost no experience developing compilers” (RFC 2). BN1's case is therefore not a counterexample but a confirmation that a peripheral participant’s contribution can remain legitimate when they enact the relational and practice-related work the community sees as essential.

Together, these cases show that newcomers, the periphery, feel this shift the hardest. AI changes the mutual engagement for their end. Engaging with the practice and the core members becomes an individual path for each contributor, rather than shared norm. The policy creation aims to overcome this by setting mutual guidelines for all members to follow equally.

The shift in mutual engagement

Read together, the positionings examined in this chapter map a single phenomenon. The strain on mutual engagement concentrates at the core-peripheral interface, where situated learning happens, and each participation level expresses a different relationship to that strain. Leaders work from the governance level to protect the conditions of practice. Core Members defend the relational mechanism from within. Peripheral Participants negotiate their entry from the threshold.

The Member/Unclassified and Regulars groups are not treated separately. Member/Unclassified positions overlap with stances already discussed across the other groups, while Regulars occupy a forum-assigned label that may overlap in practice with Core Members' engagement, and their positioning across the deliberation was conciliatory rather than directional.

The deliberation, in this reading, is not merely a community settling on a policy. It is a community working out how mutual engagement will continue to exist and sustain the practice through social interaction even as AI changes the path from newcomer to fuller participation. What the community is settling, in the end, is not solely what to do about AI but what it means to remain a community in an AI era.

Conclusion

The purpose of this study was to understand how an OSS community collectively negotiates the introduction of AI into its practice and how members at different levels of participation engage in that negotiation, through a qualitative thematic analysis of RFC threads from the LLVM discussion forum. This chapter offers a brief overview of the findings and sets out the study's contribution, limitations, and directions for future research.

Sustaining the practice under AI pressure

Across the four RFC threads, four concerns dominated the deliberation. These were *Code Understanding*, *Accountability, Reviewer Protection*, and *AI Use Disclosure*. Together they describe the conditions the community sees as necessary for situated learning to continue in the era of AI, and what the policy is trying to protect. Cross-project references to “human in the loop” suggest that this language is becoming a shared OSS repertoire under the same pressure.

Participation level shaped the position members took. Leaders spoke from a governance standpoint, holding the community together under disagreement, acting as moderators and shaping the policy in line with the community’s consensus. Core Members spoke from inside daily review work, defending review as the work through which newcomers learn the practice. Peripheral participants spoke from the threshold, negotiating their role in the community and the path toward fuller participation. Across these stances, the tension concentrated at the core-peripheral interface, where situated learning happens. AI lowers the technical barrier for newcomers but shifts the cost to Core Members, and the community's positions map onto exactly that. The policy is the response to that shift.

Contribution

Empirically, the study contributes a qualitative examination of how AI, as a technological change, can affect the sustainability of a CoP, and specifically how an OSS community handles that pressure. Existing research on AI in OSS has largely tracked the technical and legal issues that arise from AI integration (Mbizo et al., 2024; Rahman & Shihab, 2026; Stalnaker et al., 2026), with far less attention to how developers themselves discuss and respond to those changes (Feng et al., 2026; Xu et al., 2025). The study traces one community's deliberation to document those concerns and positions, offering a developer-centered view of AI integration in OSS.

Theoretically, the study applies CoP theory and LPP (Lave & Wenger, 1991) to a case where the community itself is openly debating the integration of an AI tool into its practice. In response, the community revises its shared repertoire to keep the practice reproducible. The LLVM deliberation names four conditions (*Code Understanding, Accountability, Reviewer Protection,* and *AI Use Disclosure*) under which situated learning can survive when AI enters the loop.

Additionally, the “human in the loop” term that emerged in the deliberation joins this shared repertoire as a stated precondition for the community's long-term viability under AI pressure. By analysing how members at different participation levels positioned themselves, the study further makes visible a tension in mutual engagement between core and periphery, the locus where situated learning happens, with the policy acting as a safety net to sustain the norms the practice depends on.

Three implications follow for OSS communities drafting AI policies. Reviewer time should be treated as a teaching investment rather than a quality-control cost, since reducing it without protecting review-as-teaching weakens the mechanism through which newcomers learn the practice. Policy deliberation should actively include voices from the periphery, since the strain of AI integration falls unevenly across participation levels. And policies should be framed as articulations of what the practice needs to remain reproducible, drawing on the conditions identified above. Beyond software development, any community that depends on apprenticeship-style learning faces a comparable challenge, and LLVM's framing of expert time as a teaching relationship and disclosure as a precondition for trust may help such communities articulate what their own practices require.

Limitations and Future Research

This study has several limitations that should be acknowledged. First, it is based on a single case study, the LLVM project, which limits the generalizability of the findings. While other OSS communities have also engaged with similar questions around AI-assisted contributions, the scope of a master's thesis could not allow for cross-case comparison. However, the findings may offer a useful starting point for future research examining whether similar patterns emerge in other communities.

Second, the classification of members into participation levels was based on the objective indicators from forum profiles, but also involved interpretive judgment in some cases. Members who did not clearly fit a single category were assessed based on the available evidence, and this process carries a degree of subjectivity that should be acknowledged. More broadly, qualitative analysis always reflects the interpretation of the researcher. While efforts were made to ensure transparency in the coding process through

the systematic use of a code scheme, different researchers might identify different patterns in the same data.

Finally, the study captures the deliberation process until policy adoption and not the community's experience of the policy in practice. Recent threads in the LLVM Discourse forum, such as “Concerns about low-quality PRs being merged into main” (tstellar, 2026), suggest that new concerns are already emerging in the reviewing process under the policy. A future study tracing the policy from adoption into implementation would close that feedback loop, offering a fuller account of how an AI tool policy takes shape and operates in an OSS community.

An AI policy looks on the surface, like a governance document. This thesis has shown it is something else as well. It is a community articulating, under technological pressure, what its own continuity requires. The conditions LLVM has named, including the principle of a human in the loop, are not rules for AI use alone. They are what a community has put in writing to remain itself.

Reference List

Ahuja, V., Clark, J., & WurtenBerg, J. (2025). Impact of Artificial Intelligence on Open-Source Software Development. *The International FLAIRS Conference Proceedings*, *38*. https://doi.org/10.32473/flairs.38.1.139001

Bailey, C. (2024). Artificial Intelligence Policy: What Computing Educators and Students Should Know. *Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 1*, 1–2. https://doi.org/10.1145/3649165.3699863

beanz. (2024, September 27). [RFC] LLVM project governance, September 2024 update [Online forum post]. LLVM Discussion Forums. https://discourse.llvm.org/t/rfc-llvm-project-governance-september-2024-update/81466

Bouvier, G. (2022). *Qualitative research using social media*. Routledge, Taylor & Francis Group.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Clark, T., Foster, L., Sloan, L., Bryman, A. (2021). *Bryman's social research methods* (6th ed.). Oxford University Press.

Dias, E., Meirelles, P., Castor, F., Steinmacher, I., Wiese, I., & Pinto, G. (2021). What Makes a Great Maintainer of Open Source Projects? *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 982–994. https://doi.org/10.1109/ICSE43902.2021.00093

Eghbal, N. (2020). *Working in public: The making and maintenance of open source software* (1st Ed.). Stripe Press.

Fang, Y., & Neufeld, D. (2009). Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, *25*(4), 9–50. https://doi.org/10.2753/MIS0742-1222250401

Feng, Z., Milewicz, R., Murphy-Hill, E., Menezes, T., Serebrenik, A., Steinmacher, I., & Sarma, A. (2026). Charting Uncertain Waters: A Socio-Technical Roadmap for Sustaining Open Source Communities in the Age of GenAI. *ACM Trans. Softw. Eng. Methodol.* https://doi.org/10.1145/3789210

Herring, S. C. (2004). Computer-Mediated Discourse Analysis: An Approach to Researching Online Behavior. In S. Barab, R. Kling, & J. H. Gray (Eds.), *Designing for Virtual Communities in the Service of Learning* (1st ed., pp. 338–376). Cambridge University Press. https://doi.org/10.1017/CBO9780511805080.016

JonChesterfield. (2025, September 15). *Our AI policy vs code of conduct and vs reality* [Online forum post]. LLVM Discussion Forums. https://discourse.llvm.org/t/our-ai-policy-vs-code-of-conduct-and-vs-reality/88300

Lakhani, K. R., & von Hippel, E. (2003). How open source software works: “Free” user-to-user assistance. *Research Policy*, *32*(6), 923–943. https://doi.org/10.1016/S0048-7333(02)00095-1

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation* (1st ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511815355

Li, C., & Jiao, J. (2023). LLVM Framework: Research and Applications. *2023 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 1–6. https://doi.org/10.1109/ICNC-FSKD59587.2023.10281186

LLVM Discussion Forums. (n.d.-a). *LLVM discussion forums*. https://discourse.llvm.org

LLVM Discussion Forums. (n.d.-b). *LLVM project* [Forum category]. https://discourse.llvm.org/c/llvm/5

LLVM Foundation. (n.d.). *Sponsors*. https://foundation.llvm.org/sponsors

The LLVM Compiler Infrastructure. (n.d.-a). *LLVM community code of conduct*. https://llvm.org/docs/CodeOfConduct.html

The LLVM Compiler Infrastructure. (n.d.-b). *The LLVM compiler infrastructure project*. https://llvm.org

The LLVM Compiler Infrastructure. (n.d.-c). *LLVM developer policy*. https://llvm.org/docs/DeveloperPolicy.html

The LLVM Compiler Infrastructure. (n.d.-d). *Request for comment (RFC) process*. https://llvm.org/docs/RFCProcess.html

Masaya Osaki. (2008). Work in progress - the fundamental research of cyberworlds: Potentials of open-source-style education for social revolution. *2008 38th Annual Frontiers in Education Conference*, S4A-7-S4A-8. https://doi.org/10.1109/FIE.2008.4720630

Mbizo, T., Oosterwyk, G., Tsibolane, P., & Kautondokwa, P. (2024). Cautious Optimism: The Influence of Generative AI Tools in Software Development Projects. In A. Gerber (Ed.), *South African Computer Science and Information Systems Research Trends* (pp. 361–373). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-64881-6_21

Nistor, N., Baltes, B., Dascălu, M., Mihăilă, D., Smeaton, G., & Trăuşan-Matu, Ş. (2014). Participation in virtual academic communities of practice under the influence of

technology acceptance and community factors. A learning analytics application. *Computers in Human Behavior*, *34*, 339–344. https://doi.org/10.1016/j.chb.2013.10.051

Oliveira, P., Amoakohene, D., Hocking, T., Gerosa, M., & Steinmacher, I. (2025). Governance Matters: Lessons From Restructuring the Data.Table OSS Project. *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 1–12. https://doi.org/10.1109/ICSME64153.2025.00067

Rahman, M., & Shihab, E. (2026). *Will It Survive? Deciphering the Fate of AI-Generated Code in Open Source* (arXiv:2601.16809). arXiv. https://doi.org/10.48550/arXiv.2601.16809

rnk. (2024, May 3). *RFC: Define policy on AI tool usage in contributions* [Online forum post]. LLVM Discussion Forums. https://discourse.llvm.org/t/rfc-define-policy-on-ai-tool-usage-in-contributions/78758

rnk. (2025a, December 17). *[RFC] LLVM AI tool policy: Human in the loop* [Online forum post]. LLVM Discussion Forums. https://discourse.llvm.org/t/rfc-llvm-ai-tool-policy-human-in-the-loop/89159

rnk. (2025b, October 1). *[RFC] LLVM AI tool policy: Start small, no slop* [Online forum post]. LLVM Discussion Forums. https://discourse.llvm.org/t/rfc-llvm-ai-tool-policy-start-small-no-slop/88476

Santos, I., Felizardo, K. R., Steinmacher, I., & Gerosa, M. A. (2025). Great Power Brings Great Responsibility: Personalizing Conversational AI for Diverse Problem-Solvers. *2025 IEEE/ACM 18th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)*, 93–95. https://doi.org/10.1109/CHASE66643.2025.00019

Singh, J., Gupta, A., & Kanwal, P. (2024). The vital role of community in open source software development: A framework for assessment and ranking. *Journal of Software: Evolution and Process*, *36*(7), e2643. https://doi.org/10.1002/smr.2643

Stalnaker, T., Wintersgill, N., Chaparro, O., Heymann, L. A., Penta, M. D., German, D. M., & Poshyvanyk, D. (2026). Developer Perspectives on Licensing and Copyright Issues Arising from Generative AI for Software Development. *ACM Transactions on Software Engineering and Methodology*, *35*(2), 1–39. https://doi.org/10.1145/3743133

Tan, X., Gong, Y., Huang, G., Wu, H., & Zhang, L. (2024). How to Gain Commit Rights in Modern Top Open Source Communities? *Proceedings of the ACM on Software Engineering*, *1*(FSE), 1727–1749. https://doi.org/10.1145/3660784

Tan, X., Long, X., Zhu, Y., Shi, L., Lian, X., & Zhang, L. (2025). Revolutionizing Newcomers' Onboarding Process in OSS Communities: The Future AI Mentor. *Proceedings of the ACM on Software Engineering*, *2*(FSE), 1091–1113. https://doi.org/10.1145/3715767

Treadwell, D. F., & Davis, A. (2020). *Introducing communication research: Paths of inquiry* (4th edition). SAGE Publications, Inc. WorldCat.

tstellar. (2026, February 7). *Concerns about low-quality PRs being merged into main* [Online forum post]. LLVM Discussion Forums. https://discourse.llvm.org/t/concerns-about-low-quality-prs-being-merged-into-main/89748

Wang, J., Bao, L., & Ni, C. (2023). An Empirical Study of the Apache Voting Process on Open Source Community Governance. *Proceedings of the 14th Asia-Pacific Symposium on Internetware*, 101–111. https://doi.org/10.1145/3609437.3609454

Wenger, E. (1998). Communities of practice : learning, meaning, and identity. Cambridge University Press.

Wenger-Trayner, E., & Wenger-Trayner, B. (2015). *An introduction to communities of practice: A brief overview of the concept and its uses*. https://www.wenger-trayner.com/publications/

Xiao, T., Hata, H., Treude, C., & Matsumoto, K. (2024). Generative AI for Pull Request Descriptions: Adoption, Impact, and Developer Interventions. *Proceedings of the ACM on Software Engineering*, *1*(FSE), 1043–1065. https://doi.org/10.1145/3643773

Xu, F., Medappa, P. K., Tunc, M. M., Vroegindeweij, M., & Fransoo, J. C. (2025). *AI-Assisted Programming Decreases the Productivity of Experienced Developers by Increasing the Technical Debt and Maintenance Burden* (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2510.10165

Ye, Y., & Kishida, K. (2003). Toward an understanding of the motivation Open Source Software developers. *Proceedings of the 25th International Conference on Software Engineering, ICSE '03*, 419–429.

Appendix

Table A 1

Leaders: Representative Quotes and Positioning Themes

User	Quote	Positioning Theme
RFC 1		
L1	"To be clear, I'm essentially proposing the opposite, a liberal policy on accepting such contributions, as long as they are not clear reproductions of copyrighted material... Regarding the code quality, whether it's bad or good doesn't feel like a sufficiently strong motivation to ban such tools outright, unless it becomes a spam concern."	Liberal framing, agenda-setter
L4	"As someone who does a lot of code reviews, I am not worried if someone uses AI to help them word a commit message or comment because of language barrier issues, or to help them make a two-line change in a PR. However, I care deeply if someone uses an AI to generate large portions of a sizeable PR because that really impacts how I have to approach the review... it is a significant problem (worthy of a ban) in my opinion."	Nuanced position, reviewer experience
L5	"I might agree with you until this sentence: it's not clear to me why there is more burden to you. You point to an added risk from the use of AI for contributors, but why would it somehow become your responsibility as a reviewer to look into whether the author of the patch isn't infringing on someone's copyright? As far as I know it isn't the case right now and AI does not change this."	Core-to-core disagreement
RFC 2		
L1	"Since this thread started, I filled out the LLVM dev meeting roundtable form and proposed a round table on this topic."	Deliberation manager
L9	"Yeah, working with newcomers is going to be, in the immediate sense net-negative - but the hope is that some number of these newcomers become more involved in the project and may eventually become maintainers or at least valuable contributors. This is vital for the health of the project long term."	Sustainability framing
RFC 3		
L6	"I personally have very little concern with a well-established contributor using AI assisted code generation... The same is not true for a new contributor…"	Experience-based distinction
L6	"I think the core problem here is basically this: I believe that our AI policy should be setting people up for success. Making the policy too liberal, by essentially saying 'use AI however you like as long as you own the result' is setting them up for failure."	Stricter policy advocacy
L4	"My personal position…is to have a blanket ban on use of AI. AI has already been extractive in terms of limited reviewer and triage resources in the community, and I don't think it provides sufficient value to be worth encouraging its use."	Blanket ban, persistent dissent
RFC 4		
L1	"The current draft proposal focuses on the idea of requiring a human in the loop … using LLM assistance to increase their productivity, and I really do want to enable them to do so, while also making sure we give maintainers a useful policy tool for pushing back against unwanted contributions."	Synthesizer, human in the loop
L11	"This RFC was discussed at the Project Council meeting... The proposal is now accepted."	Formal closure

Table A 2

Regulars: Representative Quotes and Positioning Themes

User	Quote	Positioning Theme
RFC 1		
R1	"I think the people that might rely heavily on AI tools are people that are just starting out with programming, so aiming at them is probably a good idea."	Newcomers as primary AI users
RFC 2		
R2	"If someone has taken time to study code, stands by it as if it were their own, and will personally engage in the back-and-forth review process then I don't mind if they wrote it from scratch, used sed, or had a room of monkeys with typewriters write a billion drafts until one worked."	Accountability over tool origin
RFC 3		
R3	"I'm sorry for the experience you had... I hope you're not entirely burned out by this situation"	Empathy toward peripheral participant
RFC 4		
R3	"I think the problem is that reviewers struggle with a wave of LLM output coming as contributions, where contributors don't have enough understanding of their PRs."	Core problem framing, understanding emphasis

Table A 3

Member/Unclassified: Representative Quotes and Positioning Themes

User	Quote	Positioning Theme
RFC 1		
MU2	"I think knowing to what extent active contributors already use AI would be good, to help guide a pragmatic policy."	Empirical grounding, pragmatic framing
RFC 2		
MU7	"Consider how it would have been done before AI was invented: one not confident enough to contribute code to LLVM would find an issue with LLVM, potentially do a little trial and error to identify the root cause, and open an issue about it."	Pre-AI contribution model, alternative framing
RFC 3		
MU11	"If a person clearly put a lot of effort into making a high-quality contribution, it shouldn't be rejected merely because it's lots of effort to review and probably not widely used."	Quality over effort, defense of good-faith contributions
RFC 4		
MU10	"We don't need more contributors who aren't programmers to contribute code. That may suck for those people who want to feel good by having some code accepted into a prestigious project like LLVM for their résumé without learning compilers, but I couldn't care less about those people. Those people can take their contributions to one of the other million OSS projects, LLVM will be just fine without them."	Exclusionary stance

Table A 4

Core Members: Representative Quotes and Positioning Themes

User	Quote	Positioning Theme
RFC 1		
MR3	"I would find it fairly insulting as a reviewer to learn that I'm the first human to attempt reading someone's code."	Disclosure as reviewer respect
MR5	"I'm guessing that people experienced with LLVM will be less likely to use AI tools than people who are new to it."	Experience-based distinction
RFC 2		
MR6	"I do agree that we should focus on the quality of contributions and not the tool used to generate them."	Quality over origin
MR9	"LLVM's policies prioritize encouraging submissions over protecting reviewer bandwidth... I favor a no-AI-by-default policy."	Restrictive stance, bandwidth priority
MR10	"... If you do not have a general understanding of the code, you are submitting and are not interested in obtaining one, please do not open PRs/issues."	Extractive contributions, accountability demand
ME1	"We should consider changing track, either to reject newcomers or to reject AI contributions. If we embrace both, our reviewers are going to burn out in rather short order. My preference would be to reject the AI contributions."	Anti-AI stance, burnout concern
ME2	"I would prefer to see LLVM adopt a no-AI policy on quality grounds."	Anti-AI stance, quality grounds
ME3	"I fall in the anti-AI camp myself and am of the opinion that this AI thing is a fad that will mostly die out in a few years."	Anti-AI stance, dismissive of AI longevity
RFC 3		
MR1	"If you are responsible for understanding what you submit then most of the issue should not be."	Accountability as sufficient safeguard
MR6	"Reviewers should feel empowered to reject reviewing a PR not specifically because it's AI-generated, but because people can't understand it."	Understanding as criterion
MR10	"This model, to me, also seems like exactly what we are trying to avoid by introducing new policy: spending valuable maintainer time reviewing patches where the contributor has not taken ownership for the patch by understanding it to a reasonable degree."	Ownership and understanding emphasis
ME2	"I strongly oppose any policy that frames machine-generated contributions as encouraged practices"	Persistent dissent, anti-permissive framing
ME2	"There is a huge number of potential improvements that we don't accept every day for a variety of reasons... Allowing AI contributions because they might contain useful fixes does not hold water."	Challenge to usefulness argument
RFC 4		
MR1	"It feels very hard to push back on this because it feels to some degree subjective (but I know it when I see it) but if a serious amounts of reviews became that much more verbose it would be a large cost in reviewer time."	Reviewer burden, subjective quality concern
MR14	"I find it really awkward to ask contributors whether they are using AI."	Disclosure discomfort, practical challenge
ME4	"In open source we generally don't judge contributions by a contributor's résumé or claimed experience. But interpersonal trust and publicly earned merit are genuinely important in open-source communities. That's not because people are biased; it's because, as imperfect as it may be, it's the lowest-cost way to make decisions at scale."	Merit-based community argument

Table A 5

Peripheral Participants: Representative Quotes and Positioning Themes

User	Quote	Positioning Theme
RFC 2		
MN2	“There are PC enthusiast users like me that are willing to contribute time and effort to the best of their abilities but lack the programming experience and intimate knowledge of the LLVM code base. The question comes down to: Are people like me more of a burden or could they bring value to the project? ...The bar of "generally understand the code they are submitting" is understandable, but that would exclude people like me.”	Peripheral self-identification, inclusion concern
BN1	“I'd say the main reason the code is bad quality is because I wrote it less than 3 months after graduating and with almost no experience developing compilers.”	Accountability defense, newcomer transparency
BN1	“I'm not writing a PR with AI and chucking it over the fence. I spent days reviewing and planning the code before I even uploaded something to GitHub.”	Active ownership of AI-assisted contribution
RFC 3		
MN2	“Any newcomer means automatically more work from the reviewer-side. This would make almost all contributions from non-experts "extractive" as they are highly likely a net-negative from the opportunity cost of view.”	Challenge to extractive framing
RFC 4		
MN2	“Instead of shutting the door for non-programmers with such language, I propose hard objective criteria to act as the AI quality filter, e.g. measurable and reproducible improvements (performance numbers, crash fixes etc.) that need to be explicitly mentioned within the MR/issue.”	Demand for objective criteria
RFC 1		
MN1	“None of the developers I know use even copilot. Apparently, it's mostly newer developers using AI but I'd certainly consider myself to be new having only started full-time within the past year. I've had only negative experiences with using AIs for code, but I haven't tried in a while.”	Newcomer perspective, negative AI experience
B1	“Maybe we can require authors to declare use of AI-generated content in the commit message using some fixed tags.”	Practical disclosure proposal
B2	“I am of the opinion that accepting any contributions involving generative ML is rather risky for the time being.”	Cautious restrictive stance
B3	“I'm in favor of allowing AI assisted contributions with the caveat that normal review and contributions standards apply.”	Permissive stance, existing standards sufficient