Mining a Decade of Event Impacts on Contributor Dynamics in Ethereum: A Longitudinal Study

M. Vaccargiu[1,5], S. Aufiero[2], C. Ba[3], S. Bartolucci[2], R. Clegg[3], D. Graziotin[4], R. Neykova[5], R. Tonelli[1], G. Destefanis[5]

[1]University of Cagliari, Italy
{matteo.vaccargiu,roberto.tonelli}@unica.it
[2]University College London, UK
{sabrina.aufiero.22, silvia.bartolucci}@ucl.ac.uk
[3]Queen Mary University of London, UK
{c.ba,r.clegg}@qmul.ac.uk
[4]University of Hohenheim, Germany
graziotin@uni-hohenheim.de
[5]Brunel University of London, UK
{rumyana.neykova,giuseppe.destefanis}@brunel.ac.uk

Abstract—We analyze developer activity across 10 major Ethereum repositories (totaling 129884 commits, 40550 issues) spanning 10 years to examine how events such as technical upgrades, market events, and community decisions impact development. Through statistical, survival, and network analyses, we find that technical events prompt increased activity before the event, followed by reduced commit rates afterwards, whereas market events lead to more reactive development. Core infrastructure repositories like Go-Ethereum exhibit faster issue resolution compared to developer tools, and technical events enhance core team collaboration. Our findings show how different types of events shape development dynamics, offering insights for project managers and developers in maintaining development momentum through major transitions. This work contributes to understanding the resilience of development communities and their adaptation to ecosystem changes.

I. INTRODUCTION

Open-source software (OSS) projects play an important role in the development and maintenance of numerous technological infrastructures. Understanding the dynamics of these projects, particularly during significant events, can provide insights into their resilience, adaptability, and overall health [1]. In recent years, blockchain technologies have emerged as a new frontier for OSS, presenting unique challenges and opportunities for collaborative development that extend and redefine traditional OSS paradigms [2]. Blockchain's convergence with OSS principles has created a distinct ecosystem with several notable characteristics. Unlike traditional OSS projects often led by a – possibly distributed – core team or a foundation [3], blockchain projects adopt decentralized governance models [4], aligning with their underlying philosophy but introducing new complexities in decision-making. Many projects incorporate native cryptocurrencies or tokens, introducing direct economic incentives for contributors and significantly impacting contribution patterns [5].Consensus-critical aspects of blockchain development are further impacted by complex technical challenges while managing community expectations,

security concerns, and regulatory pressures [6], [7]. Major events – such as security incidents, protocol changes, political attention, or even ordinary social media backlash are amplified by intense public scrutiny and can dramatically impact the project's trajectory [8], [9]. Ethereum[1] serves as an ideal case study for this research due to its position as the leading smart contract platform, its significant market capitalization, and its history of major network upgrades and community-driven changes. The project's scale, complexity, and diverse ecosystem of developers and users provide a rich environment for examining OSS dynamics in the blockchain context. Ethereum has faced numerous of the above-mentioned events throughout its history, which have tested the resilience, adaptability, and cohesion of its development community and have shaped its development path [12]. The dynamics and the effects of these events have, to the best of our knowledge, not been studied and understood.

In our study, we focus on analyzing the impact of major Ethereum events[2] on the Ethereum ecosystem by examining a set of ten key repositories. These repositories cover different aspects of Ethereum's infrastructure, including core client implementations, smart contract languages, development tools, and libraries that enable interaction with the blockchain.

---

[1]Ethereum is a decentralized, open-source blockchain platform that enables the creation of smart contracts, decentralized applications (dApps) [10], [11], generate and exchange NFTs (non-fungible tokens), and many others. Ethereum's codebase is publicly available under open-source licenses, encouraging global collaboration and modification by a diverse community of developers.

[2]A major Ethereum event represents a significant milestone or occurrence that fundamentally impacts matters belonging to at least two of the following categories: (1) ecosystem infrastructure (encompassing protocol-level changes and technical modifications); (2) market or community (including significant market reactions and community-driven changes); (3) development trajectory (covering coordinated development efforts and ecosystem-wide adaptations). Changes that affect only a single aspect of the ecosystem or require limited coordination across the network (e.g., minor bug fixes) are not classified as major. See Sec. IV for more details.

Together, these repositories form the backbone of Ethereum's ecosystem, with interdependencies that reflect the interconnected nature of the platform. Studying them provides a view of how different components of the ecosystem respond to major events, offering insights into the overall adaptability and resilience of Ethereum's OSS community.

The nature resulting from the convergence of OSS and blockchain technologies drives us to investigate the broader impacts of ten of the major Ethereum events (described in greater detail in Sec. IV). By examining how these events influence team organization, developer response rates, commit activities [13], and overall project health across key repositories, we aim to study how the Ethereum community adapts to these changes and provide insights into the ecosystem's development dynamics [14]. Among these metrics, commit activity serves as an important indicator, referring to the frequency of commits made by developers to a repository, capturing the level of ongoing development work and engagement within the project. Understanding commit activity helps identify periods of heightened or reduced activity, revealing how teams respond to major events, such as protocol upgrades or security incidents, and informing decisions about resource allocation and workload planning [15].

In our study, we pose the following research questions:

RQ 1: What is the time window during which the effects of major Ethereum events on developer collaboration in the Ethereum ecosystem are observable, before the network of collaboration returns to its normal state? Since our goal is to understand how specific major events have impacted the Ethereum developer collaboration ecosystem, it is essential to clearly define the *before* and *after* of each event, determining the time window beyond which the effects of the event are no longer observable. Through temporal analysis, we empirically established a 90-day window as the typical duration for event impacts to manifest and stabilize.

RQ 2. How do major Ethereum events impact commit activity across key repositories in the Ethereum ecosystem? By analyzing fluctuations in commit activity around major events, we can identify when developers are most actively engaged or when contributions decrease. This understanding helps reveal how development efforts are coordinated, whether on a planned or voluntary basis, and provides insights into how developers respond to major changes in the ecosystem [16].

RQ 3. How do major Ethereum events impact issue resolution time? Changes in issue resolution time indicate the efficiency and responsiveness of the development team during critical periods. This question helps assess the operational challenges faced by the team and whether developer actions are more structured (planned/allocated) or occur on a best-effort/volunteer basis [17].

RQ 4. How do developer collaboration networks change during major Ethereum events? Examining shifts in collaboration networks reveals how team dynamics, roles, and communication patterns adapt during major events. This question is important because it helps us understand the underlying social structure of the developer community, which supports or

constrains their collective response to changes in the ecosystem. We provide a complete replication package including all datasets, analysis scripts, results at **this link** to support reproducibility and verification [18].

II. RELATED WORK

Our study provides a blockchain-specific analysis of developers' activities and network structures. As shown in the following, previous studies have analyzed aspects of community interaction, event impact and network dynamics. Our work, on the other hand, focuses on a set of major events analysing how these positively influence the collaborative and technical aspects of Etheruem projects, a particular example of blockchain OSS.

a) Studies on OSS Development: The analysis of GitHub issues and comments in OSS projects has been explored by Mumtaz et al. [19], who introduce features like "assign issues to issue commenters" and analyze social dynamics within software teams. Our study extends this approach by focusing on blockchain-specific OSS projects, examining not just social interactions but also the technical contributions through commits, thereby providing a more holistic view of developer behavior. Similarly, Jamieson et al. [20] study the decentralized web communities by considering commits and the impact of project values on team dynamics. Our work diverges by employing network analysis techniques to scrutinize the structure of developer interactions over time, offering a novel perspective on how values and events influence OSS development processes. Santos et al. [21] focus on facilitating task selection in OSS projects through textual data analysis. Their approach to understanding developer interactions and sentiments mirrors our methodology, which also seeks to unravel the complexities of developer communication in blockchain projects. However, our study extends beyond textual analysis to incorporate statistical and network analysis techniques, providing a more multidimensional view of developer behavior.

b) OSS Blockchain-Specific Studies: Das et al. [22] conduct an empirical analysis of interactions on commits within blockchain software repositories. Our methodology differs in that we not only analyze commits but also integrate issue resolution patterns and developer network dynamics to understand the broader implications of major blockchain events. Chakraborty et al. [23] provide an overview of software engineering practices in blockchain projects through a survey approach. In contrast, our study applies quantitative analysis of actual project data to uncover how events specifically impact these practices. Ortu et al. [24] provide a comparative statistical characterization of traditional software systems versus blockchain-oriented software projects, exploring potential differences using a set of ten software metrics. Our research complements their findings by digging deeper into the blockchain-specific dynamics, especially how significant events influence these metrics within OSS development environments.

We build on their foundational work by applying a similar analytical rigor to more granular, event-driven data, enhancing

TABLE I: Overview of the selected Ethereum repositories, sorted by total activity.

Repository	Description	Commits	Issues	Comments
MetaMask	A widely used browser extension for managing Ethereum wallets and interacting with dApps	21121	11248	92676
Solidity	The main programming language for writing smart contracts on Ethereum	24617	5984	49744
Go-ethereum	The primary client for running Ethereum nodes, crucial for the core operation of the blockchain	15373	8071	56109
Chainlink	A decentralized oracle network that allows smart contracts to connect with real-world data, essential for many DeFi applications	24060	430	26184
Truffle	A development framework for compiling, deploying, and testing smart contracts	16047	2926	18546
Web3-js	Another library for blockchain interaction, widely used in building decentralized applications	4005	3888	21189
Hardhat	Contains the specifications for Ethereum 2, focusing on the shift to proof-of-stake and scalability	10650	2548	15715
OpenZeppelin	Provides a library of secure and reusable smart contracts, widely used in the ecosystem for security and development best practices	3648	1901	14359
Consensus-Specs	Contains the specifications for Ethereum 2, focusing on the shift to proof-of-stake and scalability	9685	920	8164
Ethers-js	A lightweight library for interacting with the Ethereum blockchain, popular among developers for its simplicity	678	2634	13961
Total	-	129884	40550	316647

our understanding of how specific blockchain events impact software development practices.

c) Network Analysis in Software Engineering: Network analysis has been employed to study community structures and interactions within software projects [25]–[32]. Ao et al. [33] and Said et al. [34] utilize network science to explore the community dynamics on the Ethereum blockchain. Our research builds upon these studies by specifically focusing on how major events such as market crashes or protocol upgrades affect the temporal and structural dynamics of developer networks, employing novel statistical techniques such as motif analysis and survival analysis.

d) Event Impact Analysis: Major events and their impacts on software development processes are highlighted by Treude et al. [35], who examine unusual activities in GitHub projects. Our study expands on this by systematically defining event windows and employing rigorous statistical methods to assess the impact of these events on multiple facets of OSS development in blockchain projects. Additionally, studies by Pejić et al. [36] and Kapengut et al. [37] focus on the impacts of COVID-19 and the transition to proof of stake on GitHub activities and network concentration, respectively.

III. DATASET

This study examines a set of 10 repositories from the Ethereum ecosystem. These repositories were selected because they represent various core components of the platform, including client implementations, smart contract programming, developer tools, and libraries for interacting with the blockchain. The goal is to understand how different parts of the ecosystem respond to major Ethereum events. The Selected Repositories are Go-Ethereum (Geth), MetaMask, Ethers.js, Web3.js, Truffle, Solidity, Chainlink, Hardhat, Consensus-Specs and OpenZeppelin Contracts, which are described in detail below and in Table I. The selected repositories represent different functional components within the Ethereum ecosystem, each serving distinct purposes while maintaining operational independence. While there are theoretical interdependencies in terms of functionality - for instance, developer tools like Truffle and Hardhat need to adapt to changes in Go-Ethereum specifications - the actual development processes and teams operate independently. Although Truffle has been officially sunset, as announced by ConsenSys, its historical significance and the transition towards Hardhat justify its inclusion in our analysis. The transition period from Truffle

to Hardhat provides valuable insights into how the community adapts to evolving tools, demonstrating how independent development teams respond to ecosystem-wide changes.

Updates to the Solidity programming language can impact tools like Ethers.js, Web3.js, and OpenZeppelin Contracts, which must stay compatible. Such interconnections mean changes in one area, like a protocol update, quickly ripple through and require adjustments across tools. MetaMask, a bridge between users and decentralized applications, relies on Web3.js and Ethers.js for Ethereum interactions. Changes in MetaMask or these libraries significantly impact user experience and development practices. Chainlink, providing external data to smart contracts, is also vital for projects requiring secure data feeds. These interconnections mean that changes from major Ethereum events propagate through the ecosystem, impacting various repositories and the developer community. Examining these linked repositories provides insight into how significant changes influence Ethereum's development and operation. This interconnected dataset enables a thorough analysis of Ethereum's systemic responses, including transitional phases like Truffle to Hardhat. Including repositories of varying sizes and activity levels captures a broad spectrum of responses within the Ethereum community. Larger repositories like MetaMask and Solidity represent core infrastructure, while smaller or transitioning ones, like Truffle's shift to Hardhat, demonstrate adaptability. Analyzing both stable and evolving repositories reveals how different parts of the ecosystem respond, whether by maintaining stability or adapting to new technologies.

IV. METHODOLOGY

Our systematic investigation of developer activity in the Ethereum ecosystem follows a structured approach, from detailed data collection through multiple layers of analysis, as illustrated in Fig. 1. This methodology enables a thorough examination of how major events influence development patterns across key repositories.

Starting with data collection from GitHub, shown in the left section of Fig. 1, we identified and extracted data from 10 central Ethereum repositories, focusing on core infrastructure and widely-used development tools. For each repository, we systematically collected three primary datasets: commit histories with metadata, issue tracking information (including creation and resolution timestamps), and developer comments.

The second phase is event selection. We analyzed Ethereum's history to identify 10 major events that have
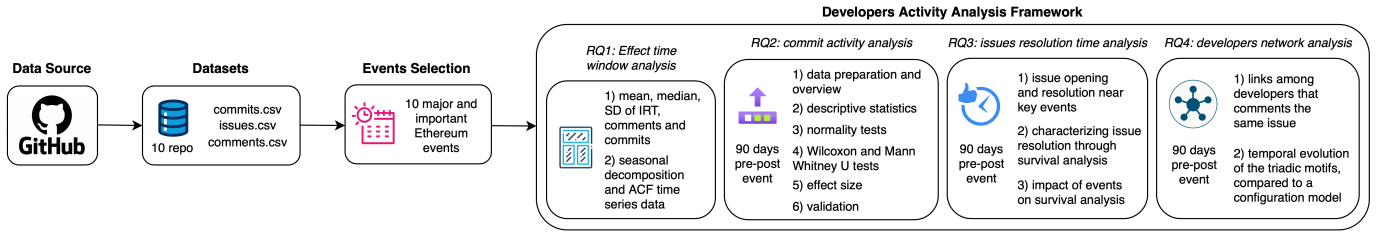
Fig. 1: Description of the methodology

TABLE II: Major Ethereum Events and Their Impact on Development Activity

Event	Date	Significance	Impact
Frontier Release	July 30, 2015	First official Ethereum release, allowing mining and dApp development.	Increased development activity and community engagement.
The DAO Creation and Hack	Jun 17, 2016	Major project built on Ethereum, hacked leading to a hard fork.	Led to Ethereum (ETH) and Ethereum Classic (ETC) split, impacting governance and developer focus.
Cryptocurrency Boom	Dec 17, 2017	Ethereum's price peak during the broader cryptocurrency surge.	Raised mainstream interest, shifting developer priorities.
Market Crash (COVID-19)	Mar 13, 2020	Ethereum's market value dropped with the global downturn.	Sparked growth in DeFi, shifting focus to decentralized finance projects.
Beacon Chain Launch	Dec 1, 2020	Introduction of the proof-of-stake consensus mechanism.	Changed development approaches and resource allocation.
London Hard Fork	Aug 5, 2021	Implemented EIP-1559, restructuring Ethereum's fee system.	Required developers to adapt to new fee structures.
Arrow Glacier Update	Dec 9, 2021	Delayed the difficulty bomb, encouraging the proof-of-stake transition.	Maintained network functionality, impacting development timelines.
Ropsten Testnet Merge	Jun 8, 2022	Key test for the mainnet transition to proof-of-stake.	Provided insights and preparation for the mainnet merge.
The Merge	Sep 15, 2022	Transition of Ethereum mainnet to proof-of-stake.	Triggered significant updates and adaptations for developers.
Shanghai Upgrade	Apr 12, 2023	Enabled staked ETH withdrawals for Ethereum 2.0.	Improved key network functionality, influencing developer activity.

shaped the ecosystem, listed in Table II. These events, spanning from Ethereum's launch to recent protocol updates, offer varied contexts for examining developer responses to distinct challenges and changes.

The identification and classification of major events followed a three-phase protocol. First, we constructed an event pool by aggregating data from authoritative sources: Ethereum Foundation communications, network upgrade histories, market data, and security incident reports during 2014-2024. Two authors then independently evaluated each event using three classification criteria. **Infrastructure Impact** related to network-wide protocol changes and technical modifications (e.g., *The Merge*, *London Hard Fork*). **Market/Community Impact** includes significant market reactions and community-driven changes (e.g., *COVID-19 crash*, *The DAO hack*). **Development Trajectory Impact** covers events requiring coordinated development efforts or establishing new patterns (e.g., *Arrow Glacier Update*, *Frontier Release*). Events meeting criteria from at least two categories were classified as major, ensuring broad ecosystem impact rather than isolated effects. For diverging classifications, a third author conducted an independent review. Event validation involved examining Ethereum development community archives (GitHub discussions, Improvement Proposals, official announcements), yielding our final set of 10 major events.

The events span from the Frontier Release (July 30, 2015) to the Shanghai Upgrade (April 12, 2023), covering technical milestones like The Merge, market events such as the COVID-19 crash, and community-driven incidents like The DAO hack. The DAO Creation which took place on April 12, 2016 and the DAO Hack which took place on June 17, 2016 are two separate but very close events (less than 90 days apart). As explicated below, we avoided overlapping effects for streamlining our analysis; therefore, we took the date of the hack as the reference date. Table II provides details on each event's significance and its impact on development activity.

The core of our analysis examines four distinct aspects of developer activity, each aligned with a specific research question.

To answer RQ1, we analyzed three key metrics across all repositories. The first metric, **Issue Resolution Time**, measures the time between issue creation and closure. The second metric, **Monthly Comment Count**, reflects the intensity of collaboration. Lastly, **Monthly Commit Count** serves as an indicator of development activity.

Based on initial observations and statistical analysis, we establish a **Pre-Event Window** covering the 90 days before each event to set baseline activity levels, and a **Post-Event Window** covering the 90 days following the event to track immediate and residual effects. For each metric, we conduct seasonal decomposition, partitioning the data into three components: **Trend**, which captures long-term movement patterns; **Seasonality**, which reflects regular, recurring fluctuations; and **Residuals**, representing remaining variation after accounting for trend and seasonality.

To validate these components, we use autocorrelation function (ACF) analysis, assessing correlations between time series values at different lags to ensure robustness in our decomposition and interpretation.

To answer RQ2, we conducted a detailed analysis of commit patterns. Our first step focused on data preparation and overview, where we normalized commit counts to enable fair comparisons across repositories with varying activity levels. Next, we performed descriptive statistical analysis, calculating the means, medians, and standard deviations of commit activity for both pre-event and post-event periods. To understand the distribution characteristics of our data, we examined histograms, Q-Q plots, and conducted Shapiro-Wilk tests [38].

For our inferential statistical analysis, we examined each repository-event pair by testing the following hypothesis: H_0: *The distribution of commit counts in the 90 days before the event is the same as the distribution in the 90 days after the event* (with H_A stating that the distribution differs). Given

that each repository serves different purposes in the Ethereum ecosystem and operates independently, we treated them as separate experiment groups. For each repository, we conducted 10 tests (one per event). Since repositories are independent, for each repository we applied the Benjamini-Hochberg procedure [39] to control the False Discovery Rate (FDR) at $\alpha = .05$ across its 10 events, which is suitable for balancing Type I and Type II errors [40] in studies akin to ours.

To quantify the magnitude of changes, we calculated effect sizes using the Z-score through the formula $r = Z/\sqrt{N}$, where Z represents the standardized test statistic and N is the number of observations. We interpreted these values using established thresholds: $> .10$ for small effects, $> .30$ for moderate effects, and $> .50$ for large effects.

Finally, to validate that our results were not due to random fluctuations, we performed a control analysis by testing against 100 randomly selected events and reshuffling time series.

For RQ3, We employed survival analysis techniques to examine issue management patterns. First, issue lifecycles are tracked by visualizing resolved issues at their closure points, while unresolved issues are observed until the study's end. To account for both resolved and unresolved (censored) issues, Kaplan-Meier estimation is used, producing survival curves that represent resolution probabilities over time. Event impacts are then assessed through log-rank tests, comparing survival distributions before and after events (against the null hypothesis: H_0**:** *H0: Events do not change how quickly issues get resolved*), with a significance level of $\alpha = .05$, and applying the Benjamini-Hochberg procedure to control the FDR for consistency with RQ2.

The final component of our methodology, for answering RQ4, examines collaboration networks. Developer interactions are represented by linking individuals who comment on the same issues, with the strength of each connection based on the number of shared issues. To observe network evolution, temporal analysis is performed over twelve-month periods segmented into three phases: a baseline phase from six to three months pre-event, an impact period spanning three months before and after the event, and a stabilization phase covering three to six months post-event.

Motif analysis identifies recurring interaction patterns, using the Configuration Model as a null model and Z-scores to quantify the significance of these patterns. Consistent 90-day windows established in RQ1 are applied throughout all analyses, ensuring comparable metrics across repositories. This methodology provides a multi-dimensional view of event impacts, examining individual contributions, issue resolution efficiency, and collaborative dynamics within the ecosystem.

Two events in our dataset—London Hard Fork/Arrow Glacier Update and Ropsten Testnet Merge/The Merge—have overlapping post-event and pre-event periods, where the three-month window of one event extends into the three-month period of another. We maintained a consistent 90-day analysis period for all events, based on the empirical findings from RQ1. Our analysis examines the relative changes in developer activity surrounding each event, with the validation against
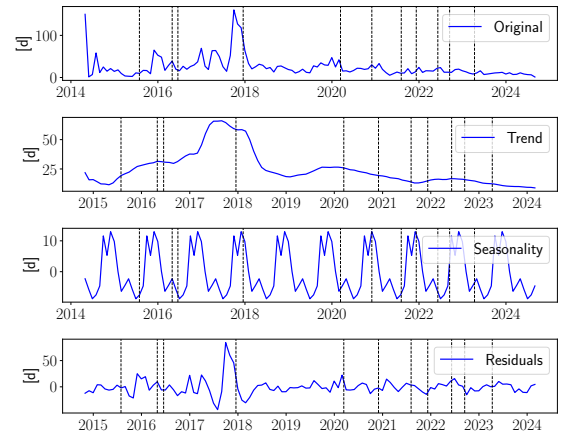
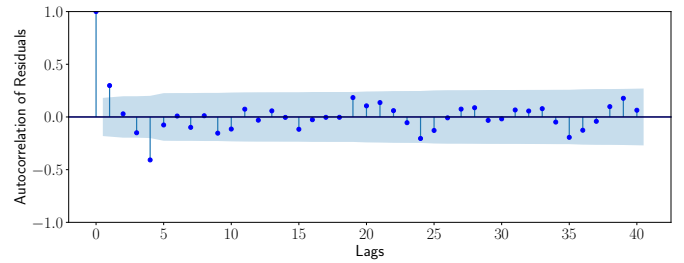Fig. 2: Analysis for the resolution times of issues

Fig. 3: ACF function for commits

random events in RQ2 confirming the observed patterns.

TABLE III: Metrics across repositories

Repository	Issue resol. time		N° of Comments		N° of Commits	
	Mean	Median	Mean	Median	Mean	Median
Overall	143.8	16.3	2473.8	2854.0	1006.8	1078.0
MetaMask	146.9	23.9	891.1	733.0	193.8	171.0
Solidity	254.6	33.2	456.4	453.0	192.3	176.5
Go-ethereum	122.5	5.4	438.3	441.0	119.2	82.0
Chainlink	85.8	12.7	323.3	220.0	293.4	278.0
Truffle	116.9	24.2	178.3	175.5	160.5	141.0
Web3-js	127.8	29.8	178.1	165.0	33.4	20.0
Hardhat	103.9	17.0	206.8	190.0	138.3	123.0
OpenZeppelin	109.8	7.3	149.6	139.0	37.6	31.0
Consensus-Specs	105.3	14.0	113.4	70.5	134.5	98.5
Ethers-js	69.1	6.0	143.9	145.0	28.2	26.0

V. ANALYZING THE TEMPORAL IMPACT OF SIGNIFICANT EVENTS ON DEVELOPER COLLABORATION

To understand the impact of significant events on developer collaboration, we define clear *before* and *after* periods for each event. This allows us to identify the time window beyond which the effects of the event are no longer noticeable.

Issue resolution time, defined as the time between an issue's creation and closure [41] ($M = 143$, $SD = 288$, $median = 16$ days). This distribution is skewed, with most issues resolved quickly and a smaller subset remaining unresolved for extended periods.

Number of comments created monthly ($M = 2473$, $SD = 1488$, $median = 2854$ comments).

Number of commits created monthly ($M = 1006$, $SD = 463$, $median = 1078$ commits)

For each of these metrics, we compute the overall mean and median values, as well as for each individual repository, as summarized in Table III. The distribution of issue resolution times is consistently skewed, with the median significantly lower than the mean across all repositories. In contrast, the number of comments and commits show closer alignment between the mean and median, indicating more balanced distributions for these metrics. To analyze the temporal patterns in these metrics, we use the median values of resolution times, comments created, and commits on a monthly basis. This approach helps account for any potential seasonality in the data. We selected the median over the mean due to the skewed distribution of resolution times, as the median provides a more robust measure of central tendency and is less affected by outliers or extreme values.

Fig. 2 presents the seasonal decomposition of issue resolution times into trend, seasonality, and residuals. The original time series (top panel) shows raw data from 2015-2024, with notable variations in median resolution times. The trend component (second panel) reveals long-term patterns, while the seasonal component (third panel) captures recurring cycles in resolution times throughout each year. The residuals (bottom panel) show the remaining variance after accounting for both trend and seasonal effects, with their random distribution suggesting the model has effectively captured the main temporal patterns. The bottom section displays the residuals, which represent the remaining variance after accounting for both trend and seasonality. The residuals reflect random fluctuations in the data that are not explained by the long-term trend or seasonal patterns. The random distribution of residuals suggests that the model has captured most of the meaningful variations. To confirm the randomness of the residuals, we compute the autocorrelation function (ACF). The ACF measures how a time series is correlated with its past values at different time lags, helping us understand the relationship between an observation at time t and earlier observations at times $t-1$, $t-2$, and so on. Ideally, the residuals should resemble white noise — random, without discernible patterns — indicating that the model (trend and seasonality) has adequately captured the time series dynamics. If the residuals exhibit significant autocorrelation, the model has not fully captured the underlying patterns, and some structure remains.

We compute the autocorrelation function (ACF) analysis for each metric - resolution times, comments, and commits. For resolution times, we observe an inverse relationship at lag 3, while comment patterns show no statistically significant autocorrelation. Fig. 3 shows the ACF analysis for commits, where bars outside the blue confidence interval indicate statistically significant temporal correlations. The analysis reveals significant lags at 1 (positive) and 4 (negative), indicating that high commit activity in one month predicts similar activity the next month, but tends to decrease after four months. The positive autocorrelation at lag 1 suggests persistent complexity, where certain issues span multiple days or indicate development

process bottlenecks. In contrast, the negative autocorrelation at lag 4 suggests that teams typically resolve longer issues or adapt processes within this timeframe, leading to decreased commit activity. Given our dataset of 129884 commits and 40550 issues, commits provide the most detailed measure of developer activity and are our primary focus. Through this ACF analysis, we define the *post-event window* as the third month after an event, when the final effects on commits are observed, with normal activity resuming by the fourth month. For consistency, we consider the 90 days before an event the *pre-event* and the 90 days immediately following as the *post-event*.

> **Answer to RQ1**: The effects of major Ethereum events on developer collaboration are observable for 90 days (approximately three months) after the event. By the fourth month, activity levels typically stabilize.

VI. IMPACT ANALYSIS OF MAJOR ETHEREUM EVENTS ON DEVELOPER COMMIT ACTIVITY

Understanding how major Ethereum events impact development activity is important for assessing how developer teams respond to network changes. By tracking commit activity before and after these events, we can determine whether development efforts are planned or reactive. This analysis highlights periods of increased or decreased engagement, offering a clearer view of how updates or disruptions affect the ecosystem's overall activity.

We observed significant variations in commit activity across repositories in response to major Ethereum events. By comparing the number of commits before and after each event, we can identify whether developers tend to be more active before or after these occurrences.

For the Frontier Release, we observed that the number of commits generally decreases post-event in Go-ethereum, Solidity, and Web3.js, while MetaMask Extension remains stable. The DAO Creation and Hack led to an increase in commits for Solidity and MetaMask Extension, indicating post-event code updates. The COVID-19 Crash resulted in more commits for Go-ethereum, Hardhat, and MetaMask Extension. Notably, Hardhat showed significant increases in commit activity after the Beacon Chain Launch, The Merge, and Shanghai Upgrade.

Fig. 4 highlights the changes in pre- and post-event activity that fall outside the gray confidence area of $r - value = .30$. The only event with a significant increase in commits is the Cryptocurrency Boom for Chainlink. Conversely, there is a notable decrease in commits for the Frontier Release in Solidity and Go-ethereum, and for the Beacon Chain Launch in Hardhat.

Fig. 5 reveals varying impacts of events on commit activity. Market-driven events (COVID-19 Crash, DAO Creation) trigger post-event commit peaks, while planned technical changes (Beacon Chain Launch, The Merge, Frontier Release) show pre-event activity concentration, suggesting proactive development. Technical milestones tend to slow subsequent commit
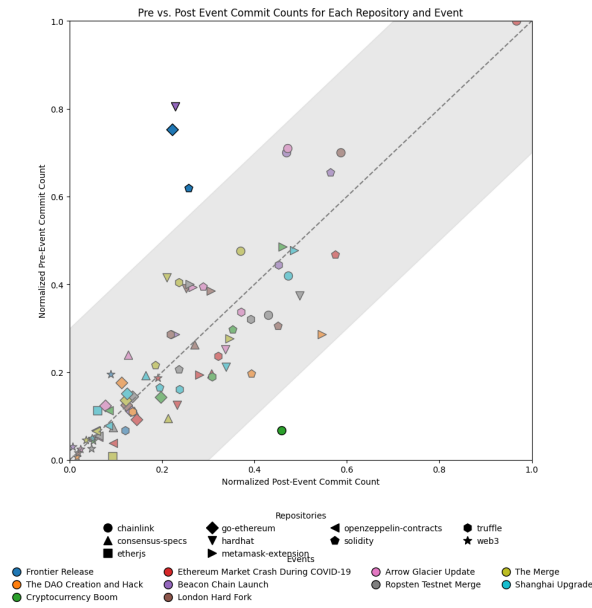
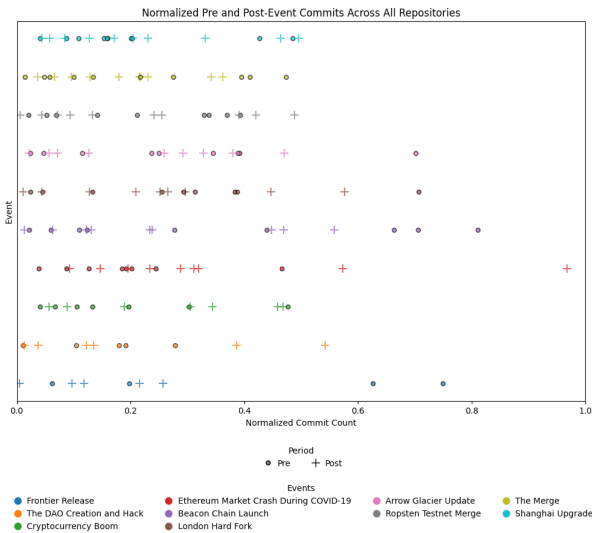Fig. 4: Pre vs. Post event commit counts for each repository and event

Fig. 5: Normalized count pre (circle) and post (cross) event commits across all repositories

activity during protocol adaptation, whereas market events stimulate reactive development responding to economic shifts. Repository responses varied by type: infrastructure repositories (Go-ethereum, Solidity) concentrate activity around core updates; development tools (Hardhat, Truffle) show post-upgrade compatibility spikes; market-sensitive repositories (Chainlink, MetaMask) demonstrate rapid responses to external pressures; and libraries (Web3.js, Ethers.js) maintain consistent activity with targeted increases during technical transitions. Shapiro-Wilk tests ($p < .05$) showed non-normal distributions for all repository-event pairs, necessitating the use of non-parametric methods.

We, hence, employed the Wilcoxon signed-rank test for our Hypothesis on comparing pre- and post-event commit activity within the same repository, as samples are paired.

Table IV shows significant results ($p < .05$) from the Wilcoxon tests after applying the Benjamini-Hochberg correction. Core repositories (Go-ethereum, Solidity) show significant changes during foundational events like the Frontier Release, while development tools (Hardhat, Truffle) display significant activity around major upgrades, indicating efforts to ensure compatibility. MetaMask exhibits heightened activity around security incidents and market crashes. We calculated effect sizes (r-values) for the Wilcoxon signed-rank test (the complete effect size values list is available in our replication package). Negative r-values were common, indicating a general decrease in commit activity after major events, suggesting a period of adjustment. The main decreases with ($r = -.86$) are for in the Web3-js repository for The DAO Creation and Hack, Beacon Chain Launch and London Hard Fork events. Further major changes are for The Merge ($r = -.83$) and Shanghai Upgrade ($r = -.82$) in Ethers-js and Frontier Release ($r = -.83$) for Go-ethereum and for solidity. Finally, market-related events such as Ethereum Market Crash During COVID-19 ($r = -.82$) and Beacon Chain Launch ($r = -.83$) have an interesting impact for OpenZeppelin.

To validate our findings, we repeated the same analysis using 100 random events and reshuffled time series for the repositories. The analysis yielded 0 significant results after correction, compared to the observed significant events in Table IV. This confirms that the observed effects are due to the selected major Ethereum events and not random chance.

> **Answer to RQ2:** Major Ethereum events significantly impact commit activity across key repositories, with the most significant events being the Frontier Release, The DAO Creation and Hack, and the Beacon Chain Launch. Core repositories show increased activity around foundational and security events, while development tools respond to major upgrades.

VII. IMPACT OF MAJOR ETHEREUM EVENTS ON ISSUE RESOLUTION TIME

To evaluate whether development efforts are planned or reactive to major events, we analyzed issue activity before and after each event. Daily issue data was collected from each repository to observe trends in issue openings and closures, focusing on shifts in resolution patterns. Kaplan-Meier estimation was used to model resolution time distributions, with event dates serving as covariates to compare resolution behavior before and after each event. Using the list of the major events from RQ2, we visualized changes in issue activity within the 1-month and 3-month timeframes identified in previous sections. Notable time-based fluctuations in issue activity across repositories highlight the importance of dynamically analyzing these patterns to capture the temporal impact of major events on resolution times.

TABLE IV: Significant Results for Wilcoxon Tests after Benjamini-Hochberg Correction ($p < .05$)

Repository	1 - Frontier	2 - DAO Hack	3 - Crypto Boom	4 - COVID Crash	5 - Beacon	6 - London	7 - Arrow Gl.	8 - Ropsten	9 - Merge	10 - Shanghai
MetaMask		x		x			x			
Solidity	x	x				x				
Go-ethereum	x			x			x			
Chainlink					x		x			
Truffle			x			x			x	
Web3.js	x				x					
Hardhat				x	x	x			x	x
OpenZeppelin										
Consensus-Specs							x		x	
Ethers.js										
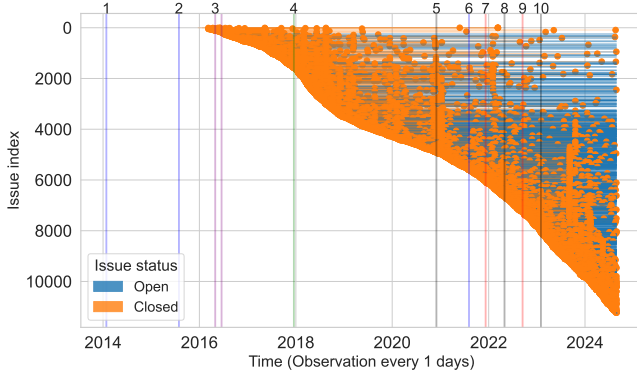
Fig. 6: MetaMask's issue resolution and events

Fig. 7: Survival analysis of issue resolution times in Go-Ethereum. The line represents the probability of issue resolution after each day since the issue was opened.
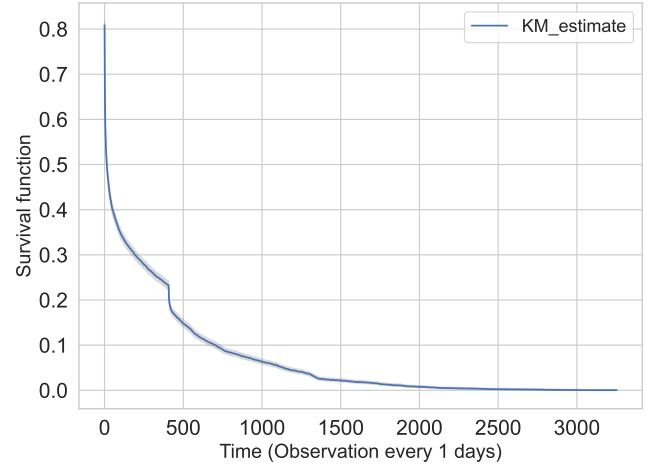
a) Issue Opening and Resolution Near Key Events: To examine how issue activity aligns with major Ethereum events, we plotted issue resolution times across all repositories, with each line representing an issue. Resolved issues are marked with an orange dot at the end, while unresolved issues extend in blue to the end of the observation period. Sorting issues by creation date allows us to track resolution patterns before and after specific events. Here, only the plot for MetaMask is shown (Fig. 6). Across repositories, a common trend is the spike in issue resolutions around major events. In MetaMask, we observe notable resolution activity around the Beacon Chain Launch (event 5) and the Arrow Glacier Update (event 7). Similar patterns are seen in other repositories. For instance, Go-Ethereum shows increased resolution activity near key events and an additional spike in older issue resolutions before 2019. Solidity also displays surges around the Beacon Chain Launch (event 5), The Merge (event 9), and the Shanghai Upgrade (event 10).

For the other repositories, similar resolution spikes are evident. For example, Web3.js, Hardhat, and Ether.js show increased resolution activity around the Shanghai Upgrade (event 10), while the Arrow Glacier Update (event 7) coincides with significant resolution efforts in Ether.js, OpenZeppelin, and Chainlink. The Merge (event 9) also influences activity in Hardhat and Chainlink. While not all resolution spikes directly correspond to major events—some relate to each repository's unique milestones—these findings illustrate a general pattern of heightened issue resolution, often in anticipation of or response to major Ethereum events.

b) Characterizing Issue Resolution Through Survival Analysis: We applied the Kaplan-Meier estimator to analyze resolution times for both open and closed issues, using the last collection date as the endpoint for unresolved issues. Fig. 7 shows the survival curve for Go-Ethereum, our most efficient repository, resolving 40% of issues within days and 75% within a year. Other repositories show varying patterns: MetaMask and Solidity resolve 50–60% within a year, while OpenZeppelin and Consensus-Specs match Go-Ethereum's efficiency. Truffle, Ethers.js, and particularly Hardhat show slower resolution times. The anomaly around 400 days in Go-Ethereum suggests systematic closure of backlogged issues.

c) Impact of Events on Survival Analysis: To assess event influence on issue resolution, we split issues into groups based on their opening time relative to each event and compared their survival curves. Fig. 8 demonstrates this approach using the Crypto Boom (event 3) in MetaMask, where diverging curves indicate event impact on resolution patterns. We used log-rank tests to determine statistical significance of these differences, applying the Benjamini-Hochberg procedure across the 10 events for each repository, consistent with our approach in RQ2. Table V presents the test results across all repositories.

While many events significantly impacted resolution times across repositories, some repositories like Solidity showed fewer significant changes, suggesting different dynamics in

TABLE V: p for log-rank test of survival curves when issues are split into two groups (before and after an event)

| Repository	1 - Frontier	2 - DAO Hack	3 - Crypto Boom	4 - COVID Crash	5 - Beacon	6 - London	7 - Arrow Gl.	8 - Ropsten	9 - Merge	10 - Shanghai
MetaMask	-	.249	< .001	.067	.173	.067	.149	< .001	< .001	< .001
Solidity	-	.721	.003	.534	.534	.534	.534	.534	.534	.534
Go-ethereum	< .001	.003	< .001	< .001	< .001	< .001	< .001	< .001	< .001	< .001
Chainlink	-	-	-	< .001	< .001	< .001	< .001	< .001	< .001	< .001
Truffle	.171	< .001	< .001	< .001	< .001	.009	.004	< .001	.002	.218
Hardhat	-	-	-	< .001	< .001	< .001	< .001	< .001	< .001	< .001
Web3-js	< .001	.310	< .001	< .001	< .001	< .001	< .001	< .001	< .001	< .001
OpenZeppelin	-	-	-	.560	.836	.764	.317	.035	< .001	< .001
Consensus-Specs	-	-	-	< .001	< .001	< .001	< .001	< .001	< .001	< .001
Ethers-js	-	-	.344	< .001	.225	.058	.007	< .001	< .001	< .001
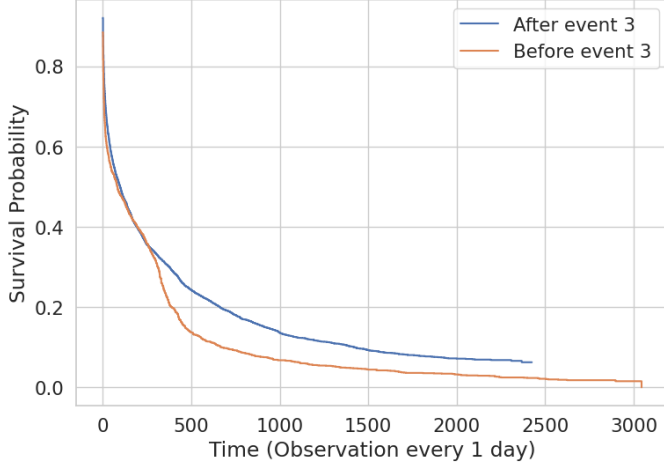
Fig. 8: Comparison of issue resolution patterns in MetaMask before and after the Crypto Boom event. The two lines representing issue resolution probabilities before and after the event, can highlight changes.

their issue resolution processes. For instance, Go-Ethereum displayed significant differences before and after all the events. MetaMask showed significant changes after the Crypto Boom and the last 3 events.

> **Answer to RQ3:** Major Ethereum events impact issue resolution times. Issue creation spikes around key events are often followed by bursts of resolutions as developers prepare for or respond to these changes. While resolution speeds vary across repositories, most issues are resolved quickly, though some repositories appear to manage the number of outstanding issues differently. Overall, issue resolution times change in a statistically significant way before and after major events for most repositories, reflecting increased activity and responsiveness. However, some repositories exhibit resilience to event effects, indicating diverse development dynamics within the ecosystem.

VIII. CHANGES IN DEVELOPER COLLABORATION NETWORKS DURING SIGNIFICANT ETHEREUM EVENTS

We constructed a network to capture interactions within the Ethereum developer community. Each node represents a contributor, and an edge represents a collaboration between two contributors who comment on the same issue. We focus on how collaboration networks change in response to events, so we do not track who created the issue, only who commented on it. Since our goal is to examine collaboration, the issue creator's role is not considered hierarchical, and the networks are undirected. Across all repositories, we identified 17829 unique comment authors, with 2774 (15%) contributing to more than one repository. Over half of these overlapping authors (1816) are concentrated in just two repositories. This indicates that each repository largely functions as a distinct network of interactions. Therefore, for this analysis, we treat each repository as an individual, self-contained network. Let us focus on MetaMask, the largest repository by total activity. We constructed an adjacency matrix where two comment authors are connected if they commented on the same issue, with the connection strength determined by the number of shared issues. This results in an undirected, weighted network of interactions among MetaMask contributors. The network, spanning from October 22, 2014, to August 28, 2024, consists of 6828 nodes (comment authors), all of whom have collaborated with at least one other contributor. The degree distribution of the network is highly right-skewed, following a long-tailed distribution. Most nodes have a low degree, meaning contributors engage with a small number of others by commenting on a few issues. However, a few nodes exhibit very high degrees, indicating involvement in numerous issues and collaboration with many contributors. This distribution is typical of a scale-free network, where a few highly connected nodes (hubs) dominate [42]. These hubs likely represent core contributors who play a key role in MetaMask's development, while the majority of contributors participate sporadically, a common pattern in open-source communities. Additional metrics further illustrate the network structure: the network density is .0027, indicating sparsity, and the average clustering coefficient is .0003, reflecting low local clustering [43]. The network contains 24 connected components, indicating some fragmentation.

Following Squartini et al. (2013) [44], we analyze the network evolution by examining topological signatures in monthly intervals from January 26, 2016 to August 2024 (104 months). This monthly resolution aligns with findings from Sec. V, where event impacts manifest within 90-day periods. We represent collaboration as symmetric, weighted networks where nodes are developers and edge weights indicate shared issue comments, yielding monthly $N \times N$ adjacency matrices A_t, with $(A_t)_{ij}$ representing interaction strength between developers i and j at month t (where $t = 1, \cdots, 104$). Beyond network size and density, we analyze higher-order

topological properties through motifs, which reveal complex interaction patterns and emerging trends. We examine the relative frequency of undirected network motifs: reciprocated dyads ($D^{\Leftrightarrow}$), open triads (V-shape, two nodes connected to a common node), and closed triads (triangles). Using the Configuration Model (CM) as a null model, we calculate Z-scores to quantify deviations from random expectations, comparing each measured quantity X to its expected value $\langle X \rangle$. Unlike the Erdős–Rényi model, CM preserves degree distribution while randomizing link weights. Following findings from Sec. VI, we analyze three key events: the COVID-19 crash (March 13, 2020), London Hard Fork (August 5, 2021), and Arrow Glacier Update (December 9, 2021). We examine Z-scores of triadic motifs, which better capture complex collaboration structures than dyadic ones, during six-month windows around each event, with grey highlighting marking the three-month impact period. As shown in Fig. 9, Go-ethereum exhibits the strongest patterns, with technical events triggering significant drops in open triads and spikes in triangles. While MetaMask and Solidity show similar but muted effects, the COVID-19 crash had minimal impact across repositories compared to technical events' substantial influence on core infrastructure collaboration patterns. The market crash in March 2020 had the least impact on the interaction structure. In contrast, the London Hard Fork and Arrow Glacier Update had more significant effects, particularly on closed triads, suggesting tighter collaboration during technical events. Go-Ethereum consistently shows higher Z-scores across both open and closed triads, indicating its role as a central hub during both financial and technical shifts, likely due to its importance in defining smart contracts. MetaMask and Geth were more reactive during market events, reflecting greater sensitivity to market conditions, while Solidity remained stable, showing its consistent role within the ecosystem.

> **Answer to RQ4:** The March 2020 market crash had the least impact on interaction structures, while the London Hard Fork and Arrow Glacier Update had a more significant effect, suggesting increased collaboration during technical events. Go-ethereum showed its central role during financial and technical changes, likely due to its importance in smart contract development. MetaMask and Go-ethereum were more reactive to market events, while Solidity remained stable, showing resilience in the ecosystem.

IX. THREATS TO VALIDITY

Construct Validity: Primary threats involve our operationalization of key concepts. Our classification of major Ethereum events involves subjectivity, though mitigated by selecting widely acknowledged events with cross-domain impact. Using commit activity and issue resolution time as development metrics may not capture all contributions or team dynamics, while network analysis through shared comments might miss other collaboration forms. We address these limita-
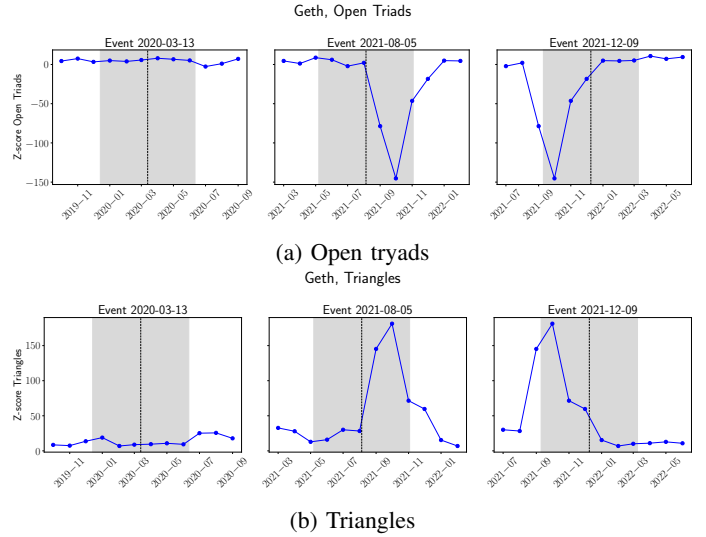

(a) Open tryads

(b) Triangles

Fig. 9: Go-ethereum repository. The black vertical line marks the event, while the grey shaded area represents the three months before and after the event. The blue line shows the trend of the Z-score for the triadic motifs over time

tions through multiple metrics and cross-repository validation. Repository selection bias is minimized by using objective criteria (activity levels, ecosystem roles) to ensure complete coverage of the Ethereum ecosystem. To ensure broad impact, we define major events as those spanning at least two categories (infrastructure, market, or development trajectory). This criterion avoids overestimating the influence of isolated events. Our classification is based on historical records beyond our dataset, including Ethereum Foundation communications, network upgrades, market shifts, and security reports from 2014–2024. While a constructed definition, it provides consistency in assessing event significance. Future work could refine this approach by exploring alternative classification methods.

Internal Validity: Several factors could affect the relationship between events and observed developer activity. First, concurrent events or changes not included in our analysis might influence our results. We addressed this through our 90-day window analysis and by validating findings against random time periods. Repository sizes vary significantly (from 678 to 24617 commits). However, our key findings and primary statistical inferences are drawn from the major repositories ($> 15K$ commits): Go-ethereum, MetaMask, and Solidity. While we analyze medium ($5 - 15K$) and smaller ($< 5K$) repositories for completeness, their results primarily serve to complement our main conclusions. This approach ensures our statistical inferences remain robust despite the size variations across the dataset.

External Validity: Our study focuses specifically on the Ethereum ecosystem during 2014–2024, examining how different types of events impact development patterns across its diverse repositories. While our findings provide insights into Ethereum's development dynamics, we acknowledge that

these patterns are shaped by Ethereum's unique characteristics as a leading smart contract platform. Factors such as its decentralized governance, token incentives, and community-driven protocol upgrades differentiate Ethereum from traditional OSS projects. Our repository selection spans multiple layers of the Ethereum ecosystem—from core infrastructure (Go-ethereum) to development tools (Hardhat, Truffle) and user-facing applications (MetaMask)—providing a broad view of development patterns within this blockchain platform. This diversity strengthens our findings regarding how different components of the Ethereum ecosystem respond to various types of events. However, we recognize that event-response patterns may differ in non-Ethereum projects, where governance models, incentive structures, and development workflows vary. Future research could explore whether similar trends hold across other blockchain and non-blockchain OSS ecosystems, particularly in projects with distinct governance mechanisms or without direct market exposure.

Conclusion Validity: We ensured statistical reliability by selecting appropriate tests based on data distributions, calculating effect sizes, and conducting multiple complementary analyses. Our survival and network analyses are based on assumptions of censoring independence and comment co-occurrence as a proxy for collaboration. To address multiple comparisons, we applied the Benjamini-Hochberg procedure to control the false discovery rate (FDR). Data completeness may be influenced by GitHub API limitations, particularly for older events. However, to support validation and reproducibility, we provide a complete replication package containing all data, code, and analysis scripts [18].

X. CONCLUSION

Our analysis of the Ethereum ecosystem offers insights into open-source software development dynamics, particularly in contexts where community-driven development intersects with market pressures. Traditional OSS projects rarely experience such direct market influences, making it challenging to study how economic forces shape development patterns. The Ethereum ecosystem, however, provides a natural experiment where market events, technical upgrades, and community decisions create conditions similar to those faced by commercial software development teams. The observed distinction between responses to planned technical changes and unexpected market events parallels challenges faced in commercial software development. Through our temporal analysis, we found that event impacts persist for approximately three months before activity patterns normalize, with key events like the Frontier Release and DAO hack showing the strongest effects. Our findings suggest that OSS communities, typically insulated from market pressures, can maintain development momentum through both planned and unplanned changes when proper coordination mechanisms exist.

Network analysis reveals how different types of events influence collaboration patterns, with technical transitions promoting stronger core team interactions and market events driving broader community engagement. This pattern, particularly evident in Go-ethereum's central role during both technical and market events, offers lessons about team structure during different types of organizational changes. Core repositories' rapid issue resolution alongside varying patterns in development tools shows OSS communities can balance market pressures with technical excellence, challenging assumptions about market versus community-driven development.

REFERENCES

[1] O. S. Malgonde, T. J. Saldanha, and S. Mithas, "Resilience in the open source software community: How pandemic and unemployment shocks influence contributions to others'and one's own projects." *MIS Quarterly*, vol. 47, no. 1, 2023.

[2] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A survey on blockchain for information systems management and security," *Information Processing and Management*, vol. 58, no. 1, 2021. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091792978&doi=10.1016%2fj.ipm.2020.102397&partnerID=40&md5=aca511db5e8bad1f04b73a12aa9a301d

[3] S. Oreg and O. Nov, "Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values," *Comput. Hum. Behav.*, vol. 24, pp. 2055–2073, 2008.

[4] Y. Liu, Q. Lu, L. Zhu, H. young Paik, and M. Staples, "A systematic literature review on blockchain governance," *J. Syst. Softw.*, vol. 197, p. 111576, 2021.

[5] M. Petryk, L. Qiu, and P. Pathak, "Impact of open-source community on cryptocurrency market price: An empirical investigation," *Journal of Management Information Systems*, vol. 40, pp. 1237 – 1270, 2023.

[6] J. Chu, Y. Zhang, and S. Chan, "The adaptive market hypothesis in the high frequency cryptocurrency market," *International Review of Financial Analysis*, 2019.

[7] E. Toufaily, T. Zalan, and S. B. Dhaou, "A framework of blockchain technology adoption: An investigation of challenges and expected value," *Information & Management*, vol. 58, no. 3, p. 103444, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378720621000185

[8] N. Iqbal, Z. Fareed, G. Wan, and F. Shahzad, "Asymmetric nexus between covid-19 outbreak in the world and cryptocurrency market," *International Review of Financial Analysis*, vol. 73, pp. 101 613 – 101 613, 2020.

[9] M. Ortu, S. Vacca, G. Destefanis, and C. Conversano, "Cryptocurrency ecosystems and social media environments: An empirical analysis through hawkes' models and natural language processing," *Machine Learning with Applications*, vol. 7, p. 100229, 2022.

[10] S. Aufiero, G. Ibba, S. Bartolucci, G. Destefanis, R. Neykova, and M. Ortu, "Dapps ecosystems: mapping the network structure of smart contract interactions," *EPJ Data Science*, vol. 13, no. 1, 2024, all Open Access, Gold Open Access. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85205696190&doi=10.1140%2fepjds%2fs13688-024-00497-8&partnerID=40&md5=115ded03111e24ef22b53f12a0650e59

[11] G. Ibba, S. Aufiero, S. Bartolucci, R. Neykova, M. Ortu, R. Tonelli, and G. Destefanis, "Mindthedapp: a toolchain for complex network-driven structural analysis of ethereum-based decentralised applications," *IEEE Access*, 2024.

[12] T. Li and X. Zhang, "Development trajectory of blockchain platforms: The role of multirole," *Information Systems Research*, 2023.

[13] M. Soto, Z. Coker, and C. L. Goues, "Analyzing the impact of social attributes on commit integration success," *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pp. 483–486, 2017.

[14] Z. Wang, Y. Feng, Y. Wang, J. A. Jones, and D. Redmiles, "Unveiling elite developers' activities in open source projects," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 29, pp. 1 – 35, 2019.

[15] Y. Tian, Y. Zhang, K.-J. Stol, L. Jiang, and H. Liu, "What makes a good commit message?" in *Proceedings of the 44th International Conference on Software Engineering*, ser. ICSE '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 2389–2401. [Online]. Available: https://doi.org/10.1145/3510003.3510205

[16] Y. Zhao, A. Serebrenik, Y. Zhou, V. Filkov, and B. Vasilescu, "The impact of continuous integration on other software development practices: A large-scale empirical study," in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2017, pp. 60–71.

[17] J. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally distributed software development," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 481–494, 2003.

[18] M. Vaccargiu, G. Destefanis, D. Graziotin, R. Neykova, S. Bartolucci, S. Aufiero, R. Tonelli, C. Ba, and R. Clegg, "Mining a Decade of Event Impacts on Contributor Dynamics in Ethereum: A Longitudinal Study - MSR 2025," 1 2025. [Online]. Available: https://figshare.com/articles/journal_contribution/_b_Mining_a_Decade_of_Event_Impacts_on_Contributor_Dynamics_in_Ethereum_A_Longitudinal_Study_b_-_MSR_2025/27629559

[19] H. Mumtaz, C. Paradis, F. Palomba, D. A. Tamburri, R. Kazman, and K. Blincoe, "A preliminary study on the assignment of github issues to issue commenters and the relationship with social smells," 2022, Conference paper, p. 61 – 65. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85133979998&doi=10.1145%2f3528579.3529181&partnerID=40&md5=469de3e90f0e156527320fb5018b7aec

[20] J. Jamieson, N. Yamashita, and E. Foong, "Predicting open source contributor turnover from value-related discussions: An analysis of github issues," in *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, 2024, pp. 678–690.

[21] F. Santos, J. Penney, J. F. Pimentel, I. Wiese, I. Steinmacher, and M. A. Gerosa, "Tell me who are you talking to and i will tell you what issues need your skills," 2023, Conference paper, p. 611 – 623. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85151379625&doi=10.1109%2fMSR59073.2023.00087&partnerID=40&md5=46ea8a7ca5630f4833d777c42c543b75

[22] A. Das, G. Uddin, and G. Ruhe, "An empirical study of blockchain repositories in github," 2022, Conference paper, p. 211 – 220. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85132445747&doi=10.1145%2f3530019.3530041&partnerID=40&md5=a89ff78af5cf85fb24a657c237be787a

[23] P. Chakraborty, R. Shahriyar, A. Iqbal, and A. Bosu, "Understanding the software development practices of blockchain projects: A survey," 2018, Conference paper. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85061489853&doi=10.1145%2f3239235.3240298&partnerID=40&md5=d55711c78c0c33daa27ae9b8dc22ed79

[24] M. Ortu, M. Orru, and G. Destefanis, "On comparing software quality metrics of traditional vs blockchain-oriented software: An empirical study," 2019, Conference paper, p. 32 – 37. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063933789&doi=10.1109%2fIWBOSE.2019.8666575&partnerID=40&md5=3d79c254596c1367059be52c2dff658d

[25] S. Bartolucci, G. Destefanis, M. Ortu, N. Uras, M. Marchesi, and R. Tonelli, "The butterfly "affect": Impact of development practices on cryptocurrency prices," *EPJ Data Science*, vol. 9, no. 1, p. 21, 2020.

[26] S. Ferretti and G. D'Angelo, "On the ethereum blockchain structure: A complex networks theory perspective," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 12, p. e5493, 2020.

[27] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 2000, pp. 163–170.

[28] M. La Morgia, A. Mei, A. M. Mongardini, and E. N. Nemmi, "A game of nfts: Characterizing nft wash trading in the ethereum blockchain," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2023, pp. 13–24.

[29] P. Louridas, D. Spinellis, and V. Vlachos, "Power laws in software," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 18, no. 1, pp. 1–26, 2008.

[30] L. Lucchini, L. Alessandretti, B. Lepri, A. Gallo, and A. Baronchelli, "From code to market: Network of developers and correlated returns of cryptocurrencies," *Science advances*, vol. 6, no. 51, p. eabd2204, 2020.

[31] A. Potanin, J. Noble, M. Frean, and R. Biddle, "Scale-free geometry in oo programs," *Communications of the ACM*, vol. 48, no. 5, pp. 99–103, 2005.

[32] S. Valverde and R. V. Solé, "Hierarchical small worlds in software architecture," *arXiv preprint cond-mat/0307278*, 2003.

[33] X. Ao, Y. Liu, Z. Qin, Y. Sun, and Q. He, "Temporal high-order proximity aware behavior analysis on ethereum," *World Wide Web*, vol. 24, pp. 1565 – 1585, 2021.

[34] A. Said, M. Janjua, S.-U. Hassan, Z. Muzammal, T. Saleem, T. Thaipisutikul, S. Tuarob, and R. Nawaz, "Detailed analysis of ethereum network on transaction behavior, community structure and link prediction," *PeerJ Computer Science*, vol. 7, 2021.

[35] C. Treude, L. Leite, and M. Aniche, "Unusual events in github repositories," *J. Syst. Softw.*, vol. 142, pp. 237–247, 2017.

[36] N. Pejić, Z. Radivojević, and M. Cvetanović, "Analyzing the impact of covid-19 on github event trends," *Sustainability*, 2023.

[37] E. Kapengut and B. Mizrach, "An event study of the ethereum transition to proof-of-stake," *SSRN Electronic Journal*, 2022.

[38] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, 1965.

[39] D. Thissen, L. Steinberg, and D. C. Kuang, "Quick and easy implementation of the benjamini-hochberg procedure for controlling the false positive rate in multiple comparisons," *Journal of Educational and Behavioral Statistics*, vol. 27, pp. 77 – 83, 2002.

[40] H. Keselman, R. A. Cribbie, and B. Holland, "Controlling the rate of type i error over a large set of statistical tests." *The British journal of mathematical and statistical psychology*, vol. 55 Pt 1, pp. 27–39, 2002.

[41] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, "Are bullies more productive? empirical study of affectiveness vs. issue fixing time," in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 2015, pp. 303–313.

[42] A.-L. Barabási, "Scale-free networks: a decade and beyond," *science*, vol. 325, no. 5939, pp. 412–413, 2009.

[43] M. E. Newman, "Communities, modules and large-scale structure in networks," *Nature physics*, vol. 8, no. 1, pp. 25–31, 2012.

[44] T. Squartini, I. Van Lelyveld, and D. Garlaschelli, "Early-warning signals of topological collapse in interbank networks," *Scientific reports*, vol. 3, no. 1, pp. 1–9, 2013.