# Mining Ethereum: Open-Source Software Development and Community Dynamics

# Abstract

Our study aims to better understand Ethereum’s open-source ecosystem based on the open-source software (OSS) theory and using data-driven analysis. Ethereum blockchain has one of the largest OSS communities with over 152,000 developers. To maintain long-run development, the OSS community has to be able to acquire and retain development talent. Leveraging existing open-source software research, we test several hypotheses that allow us to identify how development and community dynamics affect blockchain project development. We find that project awareness positively depends on the participation of peripheral developers and nonlinearly on the participation superposition of the development between project releases. In addition, the project contribution positively depends on the degree and the eigenvector centrality. Our findings provide insight into the mechanisms to maintain the sustainable development of the OSS projects.

# Introduction

The blockchain technology appeared as an open-source phenomenon and quickly developed into a major technological innovation. Among the blockchain projects the Ethereum blockchain became one of the influential projects that transformed the whole cryptocurrency industry. Founded in 2013 by couple of developers and prominent minds, the Ethereum developers’ community grew to over 150,000 developers in 2023.

In one quarter of 2022, the total number of transactions on the Ethereum blockchain amounted to 105.58 million generating revenue for the transaction validators of 834,874 ETH, or close to $1.68 billion (Alchemy 2022). In addition, the total value locked (TVL) in DeFi smart contracts running on the Ethereum blockchain was $89.5 billion in Q1 2022. Number of industry applications run on the Ethereum infrastructure, including OpenSea – the largest NFT marketplace, Tether – the largest stablecoin, and Uniswap – one of the largest decentralized cryptocurrency exchanges. Therefore, Ethereum blockchain becomes an important element in the token economy and decentralized finance and its maintenance is the important task of the open-source community. Moreover, the Ethereum Foundation has declared understanding the OSS community dynamics as one of the top priorities to build a long-term foundation for credibly neutral blockchains (Ethereum 2023).

While the prior research has established some evidence the specifics of the blockchain the prior findings may not hold true. The diversity of the project may be different: some community members may be holding more power - TBD

The blockchain and OSS model for a software development is a sociotechnical system.

The open-source project success has been defined as the number The prior OSS literature attributed open-source project’s success to a variety characteristics, including the software type, intended audience, licensing type, and organizational sponsorship.

ev,eral model specifications and different sets of control vari- intended audience, reputation of developers, licensing scoabplee,s. Moreover, the results are not just statistically meaning- and organizational sponsorship (Chengalur-Smith anfudl but also economically significant.

Sidorova 2003; Lerner and Tiróle 2005; Stewart et

# Data Collection

To conduct our analysis and test hypotheses, we use the public historical data from GitHub Archive. GitHub Archive comprises the data and content of more than 2.8 million open-source GitHub repositories and is hosted on Google BigQuery platform (Hoffa 2016). The data is stored in a hybrid database format that combines structured and unstructured data. Google BigQuery platform infrastructure allows SQL-like queries to access the entire dataset and other non-SQL requests (BigQuery 2023).

GitHub provides over 20 event types, ranging from new commits and fork events, to opening new issue tickets, commenting, and adding members to a project (GitHub 2023). We focus on activities that are most commonly used by the software developers to assess the quality and make decisions whether to join the open-source project (Dabbish et al. 2012): commits, pull requests, issues, forks, and watches. In addition, we extract the details about each event from the payload string available in the GitHub Archive database. A “payload” string field contains the JSON encoded activity description with detailed information about the content of the changes made, previous version of the file, comments the developer attached to the pull request or commit, and whether the pull request or issue were closed. To access the json-stored information, we use json\_extract() function.

Google BigQuery GitHub dataset contains data beginning 2011. However, activity archives for dates between 2/12/2011-12/31/2014 was recorded from the Timeline API that has been replaced by Events API beginning 1/1/2015. The change in API changed the structure of data and created the discrepancy in records’ schemas before and after 2015. To alleviate the discrepancy and unify the data structure for the proposed analysis, we extracted the missing information from the payload fields in tables before 2015. Due to the Ethereum project OSS development has started in late 2013, over one year of data has been extracted using the alternated query and processed to match the rest of the dataset.

In this study, we focus on the OSS activities inside the Ethereum community. The main webpage of the Ethereum software development is <https://github.com/ethereum>. The structure of the GitHub platform allow creation of different repositories, i.e., folders, within the main Ethereum webpage. The addresses of the repositories begin with <https://github.com/ethereum>/ <repository name>. Therefore, to filter data in the GitHub Archive related to Ethereum development, we simply extract all activities associated with all repositories that with <https://github.com/ethereum>/ <repository name>. Note that the GitHub functionality allows other users to create copies of the repositories and save these copied folders within their webpages. Such process is called forking and the copied repositories are referred to as forks. Though the copied repositories will have the ‘<user webpage>/ethereum/’ in the name of their repository, we do not collect data for the activities in forks.

We collect data from December 2013 – May 2023 and aggregate them at a repository –actor.login – day level. Overall, we collect 460,000 records after processing 19.08 TB of data. Using the initial dataset, we further aggregate the data at a repository – day level to test our hypotheses.

# Hypothesis Development

**Prior OSS literature has established the importance of the peripheral developer's participation for the project awareness and diffusion in the developer's community. However, prior studies did not account for the projects that have a direct market valuation, i.e., Ethereum blockchain software underlying a financial asset, and Ether price is a derivative from the Ethereum blockchain software functionality. Therefore, the nature of the project may affect the contributors’ motivation and the previously identified relationships do not hold anymore. To establish the evidence, we test this relationship using the hypothesis drawn from** Setia et al. (2012):

## 

H1: The roles of core versus peripheral developers: *Greater participation of peripheral developers in open-source projects are associated with greater awareness about open-source products*.

To test our first hypothesis, we first need to organize our dataset and construct the necessary variables. The main independent variable is the number of peripheral developers. The developers who contribute to the repository can be classified into two types: core and peripheral. The core developer is the one who is heavily involved in the project development, typically has the right to apply proposed changes to the main software file, or merge pull requests, and strategically decide the future functionality of the software product. Based on the literature, core developer has been defined as contributing more than 12% of code to the project (Mockus et al. 2002). We adopted this threshold and marked as core developers those who contributed 12% or more of total activities in a repository over the past 30 days. All other developers were marked as peripheral. We also checked for the abnormal patterns of contributions by unique contributors, e.g., one across all repositories on a daily basis, and ran our analysis without them. Typically, such activity is associated with bots and should not be considered in our analysis. Moreover, the login name of such actors contains the ‘bot’ part, and we mark them as ‘is\_bot’ in our dataset. The results with and without bot activities are robust.

As the main dependent variable, we choose different measures of the project awareness quoted in the previous literature. One measure of the project awareness is number of forks. Developers tend to fork the repositories they are interested in for two reasons: (1) they want to contribute to the development and improvement of the software, and (2) they want to use the existing code for the development of their own project. In both cases the motivation to fork is rooted in the inherent quality of the software and its recognition by developer. In addition, the technology diffusion happens by means of forks, and the wider population becomes aware of it.

Building on the prior evidence of the relationship between project popularity and structure of contribution, we point our attention to the study by Medappa and Srivastava (2019) that suggests the nonlinear relationship between the superposition of the project and its popularity. We follow with the second hypothesis:

H2: The relationship between project structure and popularity: *A moderate degree of project superposition is better for a project’s popularity than very high or very low levels of superposition*.

To test our hypothesis, we construct the measure of superposition as the main independent variable. We calculate the number of developers who contribute to the repository over a time period and divide it over a number of releases made within the same time frame. The resulting metric is distributed between 0 and 1 by definition. If degree of superposition equals exactly 1, all of the releases were implemented by individual developers and added sequentially. The degree of superposition decreases as a project adopts a concurrent development approach and approaches 0 as a greater number of individual contributions get accumulated into individual releases (versions) of the project. Since our dependent variable is *project’s popularity* and is congruent with the dependent variable in our first hypothesis, we also control for the number of core and peripheral developers in our analysis.

Our third hypothesis is drawn from Grewal et al. (2006) who are the first ones to point at the significance of the network structure for project success.

3

third

Our next hypothesis is drawn from Singh et al. (2011).

H4: The nuanced relationship between developer repeated collaboration and project success: A moderate level of internal cohesion within a project is better for a project’s success than very high or very low levels of internal cohesion.

To test our next hypothesis, we construct the measure of internal cohesion following Singh et al. (2011). The presence of repeated collaborations among project developers is related to strong interpersonal connections (Uzzi 1997). We calculate the number of developer pairs from the focal repository that worked on other repositories within Ethereum project and divide it over the total number of pairs that exist in the focal repository to calculate the number of repeated ties, or the internal cohesion metric. The resulting metric is the main independent variable. The main dependent variable is a project’s success. Subsequent to the extant literature, we consider a project’s successas a project’s rate of knowledge creation and measure it as number of commits (Boh et al. 2007, Crowston et al. 2003). We also control for the lifetime of the project since its initiation (Duration).

# Hypothesis Testing

## Hypothesis 1

To test our first hypothesis, we run a panel fixed effects regression:

(1)

In equation (1), i – is the index of a repository within the Ethereum project, t – index of a time period, is the repository fixed effect that accounts for unique attributes of the repository that are not captured by other variables, – idiosyncratic error. As a measure of product awareness (), we use the number of repository forks (Forks). The main independent variable is the number of peripheral developers (Peripheral). We also control for the number of core developers (Core) and the lifetime of the project since its initiation (Duration).

After we estimate the regression, our results confirm that the tested hypothesis: Number of peripheral developers is positively associated with the repository awareness (number of forks). Addition of one peripheral developer increases the number of repository forks by 0.205 (column 1, Table 1). The results are presented in Table 1.



**Table 1. H1 and H2 Testing Results**

|  |  |  |
| --- | --- | --- |
| Variable | (1)  H1: FE, DV=Forks | (2)  H2: FE, DV=Forks |
| Peripheral | 0.205\*\*\* | 0.206\*\*\* |
|  | (0.024) | (0.024) |
| Core | 0.003 | 0.002 |
|  | (0.002) | (0.002) |
| Superposition |  | 6.412\* |
|  |  | (2.757) |
| Superposition^2 |  | -7.903\* |
|  |  | (3.550) |
| Duration | 0.0002 | 0.0001 |
|  | (0.0002) | (0.0002) |
| Repository FE | Y | Y |
| Num obs | 1,593 | 1,593 |
| Adj R-sq | 0.644 | 0.645 |

**Note.** Heteroskedasticity-consistent and autocorrelation-corrected standard errors in parentheses; + p<0.1, \* p<0.05, \*\* p<0.01, \*\*\* p<0.001

The result in Table 1 signify that peripheral developers are the drivers of the social diffusion in the collaborative networks like OSS community. The awareness about the project, or the repository in our case, is increasing as there are more developers who show interest in it but not yet heavily involved in the development activity (less than 12% of all activities in the past 30 days). Peripheral developers are involved in many other projects on the platform, therefore, when interaction with the focal repository is reflected on their profile, their collaborators are exposed to the focal project. The attention to the focal project can be reflected by subscribing for the repository updates (watching) or forking the code for future modification or use. While watching serves as a powerful signal of popularity of the code, number of forks signify the social recognition and impact (Petryk et al. 2023). Results for number of watches are consistent with the fork effects and are omitted from the submission due to the space limits.



## Hypothesis 2

We run a panel fixed effects regression (2) to test our second hypothesis:

(2)

In equation (2), i – is the index of a repository within the Ethereum project, *t* – index of a time period when the project software release has occured, is the repository fixed effect that accounts for unique attributes of the repository that are not captured by other variables, – idiosyncratic error. include the number of peripheral developers (), the number of core developers (), and the lifetime of the project since its initiation ().



The result in column 2 in Table 1 signifies that there is a non-linear relationship between superposition and project popularity ( is significant). As the level of superposition in the project increases, the project popularity increases but to a certain point. After a certain point, the further increase of superposition the project popularity starts decreasing ( is negative). Low superposition signifies that project releases occur after large number of contributions are accumulated. Such “productive deferral” makes the project less attractive to the developers due to lack of autonomy and independence in completing complex tasks. Hence, the developers are reluctant to follow such projects (Ryan and Deci 2000). High superposition signifies the high level of work independence that satisfies the autonomy need. However, the lack of collaboration and exchange of ideas creates the negative affective state and deters the interest to the project (Medappa and Srivastava 2019). Developers need the decent amount of challenge and autonomy to be interested in contributing to the project and opportunities to work with other developers on more complex tasks (Ke and Zhang 2010).

## Hypothesis 3

We run a panel fixed effects regression (3) to test our third hypothesis:

(3)

In equation (3), *i* – is the index of a repository within the Ethereum project, *t* – index of a year, is the constant term that accounts for factors that affect the project success metrics not captured by other variables, – idiosyncratic error. As a measure of product success (), we use a metric of contribution – a number of commits (Commits). are the coefficients of interest and measure the effects of centrality measures on the project technical success. include the lifetime of the project since its initiation (). The results are presented in Table 2.

The result in column 1 in Table 2 signifies that there is a statistically significant relationship between a network structure and project success. As the degree centrality increases, the projects become more interconnected with others by the common developers that attracts more contributions (). As the eigenvector centrality increases, the projects become more central in the network that turns them into magnets with respect to more contributions (). The betweenness centrality does not have a significant effect.

**Table 2. H3 and H4 Testing Results**

|  |  |  |
| --- | --- | --- |
| Variable | (1)  H3: FE, DV= lnCommits | (2)  H4: FE, DV= lnCommits |
| Degree | 0.002\* | 0.002\*\* |
|  | (0.001) | (0.0007) |
| Eigenvector | 7.682\*\*\* | 7.666\*\*\* |
|  | (1.723) | (0.519) |
| Betweenness | -0.0003 | -0.0003 |
|  | (0.0002) | (0.0003) |
| IntCohesion |  | -0.029\* |
|  |  | (0.012) |
| IntCohesion^2 |  | 0.0003\* |
|  |  | (0.0001) |
| Duration | -0.294\*\*\* | -0.292\*\*\* |
|  | (0.028) | (0.025) |
| Repository FE | Y | Y |
| Num obs | 1,634 | 1,634 |
| Adj R-sq | 0.113 | 0.115 |

**Note.** Heteroskedasticity-consistent and autocorrelation-corrected standard errors in parentheses; + p<0.1, \* p<0.05, \*\* p<0.01, \*\*\* p<0.001



## Hypothesis 4

To test our fourth hypothesis, we run a panel fixed effects regression:

(4)

In equation (4), i – is the index of a repository within the Ethereum project, *t* – index of a year, is the repository fixed effect that accounts for unique attributes of the repository that are not captured by other variables, – idiosyncratic error. include the lifetime of the project since its initiation () and three measures of the project centrality (, , and ). The results are presented in Table 2.



The result in column 2 in Table 2 signifies that there is a statistically significant non-linear relationship between the internal cohesion and project success. This finding is opposite to the common prediction. The positive sign of the quadratic term () signifies as the internal cohesion increases, its effect on the project’s technical contribution decreases; after the turning point, the further increase in the internal cohesion increases the contribution to the project. While the moderate levels of cohesion might be beneficial for participant learning (Singh et al. 2011), the learning tendencies might not produce the highest contribution output. One possible explanation is that when the cohesion is low, the lack of trusted relationship between developers leads them to higher individual responsibility for the project resulting in higher levels of contribution. On the other hand, in highly cohesive communities the developers would be more productive as the shared common knowledge will eliminate the coordination cost and the project contributions will increase.

# Discussion

The open-source model become ubiquitous in creating independent technological innovation like blockchain and maintaining existing corporate software like Twitter (Twitter) or GPT-2 (OpenAI) and taking it to new directions. The work structures of the OSS project have been of attention by prior research, however, their impact on the project success were investigated in isolation form one another. In this study, we aim to aggregate the previous knowledge of how the organizational structure of the OSS project affects its success measures. In addition, we add the time dimension in our analysis to explore how the evolution of the collaborative network structures affects the productivity and the popularity of the project. Considering the steady shift toward FLOSS projects in producing technological innovation, like blockchain or machine learning algorithms, combining prior theoretical knowledge into a congruent theory is important for better understanding the phenomenon and developing strategies on management of the new generation of the technological innovation.

Using a big data dataset from one of the larges blockchain communities (Ethereum), we test how the existing OSS hypotheses developed over past 15 years interact with each other. We find that …

# Reference List

1. Hoffa F. 2016. GitHub on BigQuery: Analyze all the open source code. <https://cloud.google.com/blog/topics/public-datasets/github-on-bigquery-analyze-all-the-open-source-code>
2. BigQuery 2023. <https://cloud.google.com/bigquery/>
3. GitHub 2023. <https://docs.github.com/en/webhooks-and-events/webhooks/webhook-events-and-payloads>
4. Mockus, A., R. T. Fielding, J. D. Herbsleb. 2002. Two case studies ofopen source software development: Apache and Mozilla.ACMTrans. Software Engrg. Methodology11(3) 309–346.
5. Petryk, M., Qiu, L., and Pathak, P. 2023. The Impact of Open-Source Community on Cryptocurrency Market Price: An Empirical Investigation. MISQ, Forthcoming.
6. Uzzi, В. 1997. "Social Structure and Competition in Interfirm Networks: The Problem of Embeddedness," Administrative Science Quarterly (42:1), pp. 35-67.
7. Ke W, Zhang P (2010) The effects of extrinsic motivations and satisfaction in open-source software development. J. Assoc. Inform. Systems 11(12):784–808.
8. Ryan RM, Deci EL (2000) Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Amer. Psych. 55(February):68–78.
9. Crowston, K., H. Annabi, J. Howison. 2003. Defining open source software project success. Proc. Internat. Conf. Inform. Systems, ICIS, Seattle, WA, Association for Information Systems, Atlanta, GA.
10. Boh, W., Slaughter, S., and Espinosa, J. 2007. "Learning from Experience in Software Development: A Multilevel Analysis," Management Science (53:8), pp. 1

**Appendix**

**Table A1. Data Structure for Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Repository | Date | User login | Is core (1/0) | Is bot  (1/0) | Num activities |
| A | Date1 |  |  |  |  |
| … | … |  |  |  |  |
|  |  |  |  |  |  |
| A | DateN |  |  |  |  |
| B | Date1 |  |  |  |  |
| … | … |  |  |  |  |
|  |  |  |  |  |  |
| B | DateN |  |  |  |  |

**Table A2. Auxiliary Data Structure to Compute Is\_Core Developer Variable**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Row | Repository | Date | User login | Num activities | Rolling count of activities, last 30 days |
| 1 | ethereum/alethzero | 2015-08-17 | 138296 | 1 | 31 |
| 2 | ethereum/alethzero | 2015-08-17 | 138296 | 1 | 32 |
| 3 | ethereum/alethzero | 2015-08-17 | 138296 | 1 | 32 |
| 4 | ethereum/alethzero | 2015-08-17 | 138296 | 1 | 30 |
| 5 | ethereum/alethzero | 2015-08-17 | 138296 | 1 | 30 |