A CASE STUDY OF OPEN-SOURCE CRYPTO-CURRENCY DEVELOPMENT: BITCOIN

Submitted by:

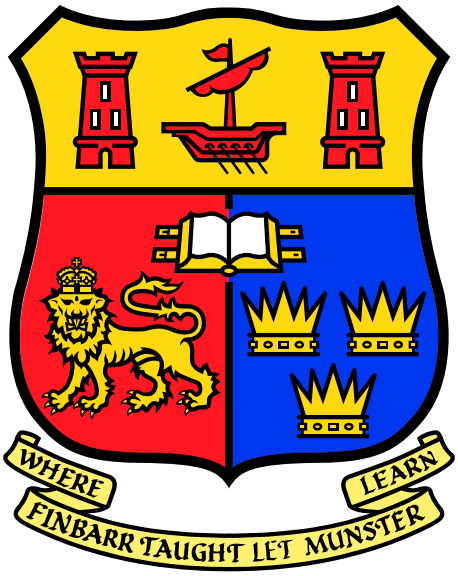
CHARLES BEYER

Supervisor:

KLAAS-JAN STOL

Second Reader:

MARC VAN DONGEN



Computer Science Department
University College, Cork

August 31, 2018

Abstract

In ten years of existence, Bitcoin proved to be the most successful cryptographic currency in the history. Originally created by Satoshi Nakamoto, first as a WhitePaper in 2008, followed quickly by the source-code of the original software in 2009. The Bitcoin software is completely open-source and is the first fully decentralized peer-to-peer version of electronic cash system that is not controlled by any central authority or control point that could be compromised. By using GitHub archive of data, we quantify aspects of developer participation, community growth and internal issues for this OSS project. We then use the price of the Bitcoin token to correlate and reveal some important patterns that may indicate new behaviors in the OSS community. We conclude that a new economy got created in the OSS community by the issuing of crypto-currency that get real-world value from its core and volunteer work.

Declaration

I confirm that, except where indicated through the proper use of citations and references, this is my original work and that I have not submitted it for any other course or degree.

Signed: 

Charles BEYER
August 31, 2018

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Bitcoin software is a decentralized cash system, with no single overseeing authority in the network [Nakamoto 2008]. This makes Bitcoin stand out from the standard fiat currencies as it is the first fully decentralized peer-to-peer version of electronic cash system.

However, Bitcoin itself can be viewed in two different aspects:

- An open-source software, originally created by Satoshi Nakamoto and developed by the community.
- A virtual currency, generated by the Bitcoin software itself and used as a speculative asset in financial markets.

In almost 10 years, the price of the Bitcoin currency increased from 0.10$ to over 20,000$ in January 2018. This price surge and popularity among the Open-source scene raises some questions in the way Bitcoin operates.

Despite being an open-source project, in which the community contribute to the project voluntarily, Bitcoin introduced a new economic incentive for developers to monetize their work by the issuing of crypto-currency that get real-world value from its core and volunteer work.

To our knowledge, no previous research has investigated the development process of Bitcoin and studied it while considering the impact of its financial aspect. Earlier studies have pointed out the economics of the competing market of Crypto-currencies [Zhang 2014], and noticed that some events in the software aspect of a crypto-currency could highly influence the price of a crypto-asset. What's more, the majority of the previous research is singularly focused on Bitcoin as a speculative asset rather than a software. An important shortcoming of previous studies is that they look at specific Bitcoin price determinants, without considering the in-depth development of the software itself and the interactions between them.

Earlier studies have pointed out the importance of doing research on crypto-currencies like Bitcoin as the technological innovations they bring affect the financial market and the Internet [Matta, Marchesi, and Lunesu 2015].

Thus, to close this research gap, this paper presents a case study of the development of the open-source crypto-currency project Bitcoin and the influence the Bitcoin's price has in its own development. This study is based on the data from GitHub, which is made available through the GhTorrent and GithubArchive projects. By using them, we can explore the whole history of the Bitcoin project and carefully analyze and identify some important factors that may affect other crypto-currency projects in the future based on the data we gathered from Bitcoin.

1.1 Research Questions

The goal of this work is to obtain a deep understanding of the whole development process of Bitcoin and to analyze the impact of the price factor on the project continuity. Specifically, the research questions are:

RQ1: *What is the process used to develop Bitcoin?*

The Bitcoin project is an open source software and based on its history and the philosophy of its creator, Bitcoin should be both developed and used in a decentralized manner. To answer this question, we construct a brief description of the Bitcoin development.

RQ2: *How many people are working on the Bitcoin software?*

We want to know how large the Bitcoin development community is, and identify the core members of the project. Through this review, we will be able to confirm if the development of Bitcoin is indeed decentralized..

RQ3: *Is the Bitcoin open-source software an active community project?*

We want to evaluate the project health within the Bitcoin development community. We will use three key indicators; the number of contributions made in the project, the community activity and their interactions and finally the pull-requests merge rate. Using these, we are able to describe the behavior of the Bitcoin project and review the activity of the latter and its consistency.

RQ4: *Did the introduction of the price factor have an impact on the continuity of its development?*

Bitcoin can be viewed as two things; both as a complex software and a digital currency. Due to the lack of regulations and the price volatility of this currency in free-markets, we analyze since the beginning of its development if the price-induced a change of behavior in the Bitcoin community.

RQ5: *What is the relation between the price and the project forks?*

Our second concern with the price of Bitcoin is its impact on the Bitcoin forks behavior. Forks can have two purposes; be used as a personal environment to work on the project and merge afterward the changes to the main branch or to create a copy of the project

in order to start independent development and go on another direction. The latter often have a negative impact on the project as it implies a conflict within the developer community and the only solution is to divide it in order to go separate ways. Similarly to the previous question, we examine the number of forks made over time and compare it to the price of Bitcoin.

This paper address key questions about the Bitcoin development process, the interaction of its developers and community, and the results of its financial value.

In Chapter 2, I summarize previous research made on the Bitcoin software and the state of open-source development. I provide in Chapter 3 a detailed description of the methods I used to retrieve the data that serve this study. Findings are presented in Chapter 4 and the implications of these findings are discussed in Chapter 5. Finally, I present my conclusions in Chapter 6.

Chapter 2

Background and Related Works

2.1 The Bitcoin currency

In ten years of existence, Bitcoin proved to be the most successful cryptographic currency in the history. Originally created by Satoshi Nakamoto, first as a WhitePaper in 2008, followed quickly by the source-code of the original software in 2009.

The Bitcoin software is completely open-source and is the first fully decentralized peer-to-peer version of electronic cash system that is not controlled by any central authority or control point that could be compromised [Nakamoto 2008]. Bitcoin is not the first cryptographic currency, and throughout the 1990s, many variations and schemes were researched and proposed. Even if early versions of cryptography currencies were properly implemented, they all had the disadvantage of being on some aspects centralized and consequently vulnerable. Cryptography open-source projects helped to develop and to improve at a quick pace security on algorithms and protocol works which would not have been obtained otherwise on proprietary works. This leads further down the line, to the creation of the open-source project Bitcoin, implementing and reworking a lot of these early researches and works.

Since its introduction in financial markets, Bitcoin has emerged as a fascinating phenomenon; without any central authority issuing the token, it became a potential alternative currency to the standard fiat currencies, with low fees, a transparent algorithm, and the ability to visualize transactions inside a decentralized ledger. The surge of interest and increase of the value of Bitcoin may be fueled by the lack of confidence in the usual banking system and its lack of transparency [Bornholdt and Sneppen 2014].

Instead of having a bank account, managed by a central authority, each user generates a Bitcoin address, that consists of a public key and a private key. Every Bitcoin is associated with the public key of the owner and payments are signed by the owner using its private key. To avoid any fraud in the Bitcoin network, the participants must agree on a single valid transaction history. This process, also called mining, is designed to be computationally difficult. The only way to hijack the Bitcoin system is by possessing the majority of the computational power of participating parties (also called a 51% attack).

Therefore, the more participants in the network, the more secure it is as more resources will be allocated to the validation process. To provide an incentive in this mining process, Bitcoins are created periodically and distributed among the nodes participating in this computation process.

The only way to obtain Bitcoin is by either mining it or trading it with another user who possesses some; the price of Bitcoin itself is determined solely by the market and the demand. It seems that it is ambiguous to define crypto-currency as either fiat money or commodity money, but it is certain that it was originally designed to be used as a medium of exchange.

Kristoufek [2013] tried to understand the price volatility of Bitcoin by measuring search queries on Google and keywords on Wikipedia. He concluded that the bubble and bust cycles of can be partially explained by the high interest in the currency. This bubble and bust cycle phenomenon is also called a "bull-run" and happened multiple times during the lifespan of Bitcoin.

Polasik et al. [2014] did a similar research by using Google search results in order to analyze the price fluctuation of Bitcoin. They concluded that the Bitcoin price formation is the result of both its popularity and the transactional needs of its users; as the price and demand increased, it caught the attention of the media and the word-to-mouth which introduced even more people to the project. This subsequently created an effect that surged a spike of interest, where new users would search on the internet to gain more information.

Kondor et al. [2014] identified two distinct phases in the lifetime of Bitcoin; first when the system was new, and no business accepted Bitcoin. Bitcoin was more or less seen as an experiment rather than a real currency. However, once Bitcoin received wider public attention, its financial value increased and the Bitcoin network started to be used as a real currency.

Despite Bitcoin being an open-source project, no research has been made focusing in its open-source development process. Apart from the work cited before, to the best of our knowledge, no previous work has studied Bitcoin's OSS development process in details. Most of the works we could find focus deeply on the economical aspect of the Bitcoin currency, but its software aspect deserves to be analyzed as well.

2.1.1 The alternative crypto-currencies

Bitcoin can nowadays be used as a starting point for creating digital currencies [Pirjan et al. 2013]. These decentralized alternative currencies use the same basic functional blocks as Bitcoin to implement digital currency functions. Because the development of Bitcoin has been conservative and value-preserving, by avoiding the introduction of errors and the longevity of the network, people will focus instead on alternative coins that do not have the restrictions of a production system like Bitcoin [Franco 2014]. Furthermore, because it is so easy to fork Bitcoin, many alternative coins are not significantly different from the Bitcoin digital currency. Most of the time, these forks are not useful, however, there are notable versions that bring significant improvements and

innovations from the original model. These improvements can be categorized in three directions:

- *New monetary policy*: Improving the method of distribution of wealth.
- *Innovative consensus mechanism*: Improving on the trust, traceability and immutability of the network.
- *Specific improved features*: Adding more functionalities such as anonymity to the network.

From a historical point of view, the very first alternative currency made through the forking process of Bitcoin is IXCoin, launched in 2011. This currency changed many parameters originating from Bitcoin, for instance, it would increase the generation rate of new tokens. This digital currency was also the first crypto-currency to implement a new Proof-of-Work consensus called Scrypt meant to prevent the denial of service attacks. Another early alt-coin successful nowadays is Litecoin; it implements the same Proof-of-Work consensus as IXCoin but with more efficient generating block methods. The alt-coin development continued during 2011 to 2012 however, starting at the year 2013, the alternative coin development increased exponentially, with new projects no longer forking the Bitcoin codebase itself but forking instead the already well-developed codebase of other alternative coins. The Figure 2.1 shows the hierarchical view of forked projects originating from Bitcoin. The crypto-currencies with an orange filled-circle are projects that are still active today. On the other hand, crypto-currencies with a non-filled circle are defunct projects. Out of 105 retrieved projects, over 55 of them are now defunct, which indicates that more than half of these created crypto-currencies are no longer active.

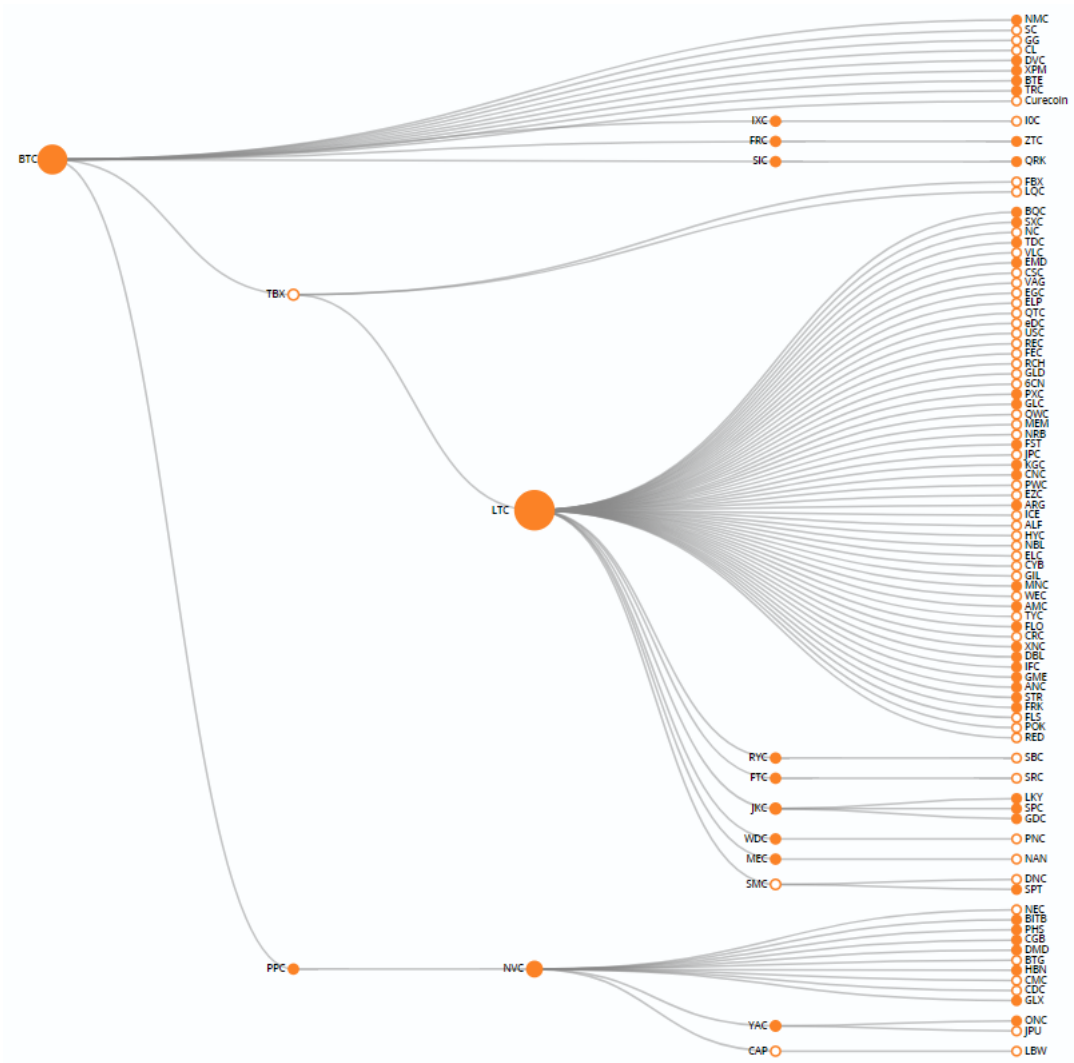


Figure 2.1: Running coins forked from Bitcoin in July of 2013 [*Map Of Coins*]

As the cost of launching a new currency is very low and fairly easy, the number of released alt-coins became very high. Nowadays, this number increased exponentially with over 10.000 new crypto-currencies. the logic behind creating its own crypto-currency is to bring significant improvements compared to the Bitcoin project to increase the value and efficiency of the blockchain technology. On the other hand, the Bitcoin network, even though it seems deprecated compared to other alternative currencies, presents both direct and indirect network effects; the number of people that uses Bitcoin increases, its value increases as well and as the number of people that spends Bitcoin increases, the support of the network becomes bigger.

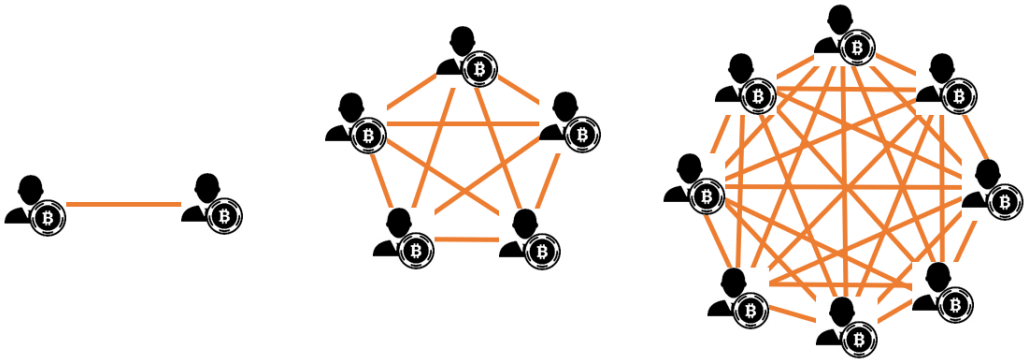


Figure 2.2: Example of a Network effect with Bitcoin users

In fact, if a network has n users, then the network will grow proportionally to n^2. With liquidity of an asset being the degree in which an asset can be sold without impacting its market price, it could also become an economic measure of its network effect. This network effect can have a huge impact on the sustainability of a crypto-currency, What's more, for the Bitcoin to be overthrown or replaced by another alternative coin, it would need to transition the network effect from currency to another which would prove to be difficult due to coordination problems.

Farell [2015] stated that "*the crypto-currency industry is rapidly moving forward*", and despite major negative events such as the Bitcoin wallet theft of Mt. Gox in 2014, government shutdowns and the drastic decrease in value, the industry has shown itself to be resilient. What's more, the industry has shown its strength by having new crypto-currencies bringing workable solutions to deficiencies in the blockchain technology with a high market valuation, indicating a strong interest by the broader audience.

Bornholdt and Sneppen [2014] considers the rise of Bitcoin to the status of currency due to the fact that it can now be used to buy real products and services on the World Wide Web. This emphasizes the fact that money is a social concept that can evolve from the contact between people. Bornholdt and Sneppen [2014] also determined with a voter-like dynamics that crypto-currencies are at their core inherently interchangeable and that Bitcoin holds no special advantages over other crypto-currencies.

2.2 The state of Open-source development

In Open-source development, the software is made publicly available and shared through the internet [Hertel, Niedner, and Herrmann 2003]. Every developer with sufficient skills can join an open-source project and participate in its development. Usually, developers contribute to an open-source project without asking for something in return. Open-source development goes back to software development itself; in the 1960s, researchers had to share software code and work in open sharing because com-

mercials solutions were not available yet [Moon and Sproull 2002]. Afterward, when commercial software development increased, open-source development became an attractive alternative since it allowed users to modify and improve software according to their own needs without any restriction.

Stallman [1994], one of the first public advocates of the open-source movement, introduced a political aspect in open-source in which society "*needs information that is truly available to its citizens*". In fact, people must be able to read, fix, adapt and improve software and not just operate them.

According to Weber [2004], the creation of a particular kind of software can be seen as an "*experiment in social organization around a distinctive notion of property*". Open-source is in fact fundamentally based around the right to distribute and not the right to exclude. Why do people work on free software such as Bitcoin? According to Young et al. [1999], "*this motivation is not just altruism*". This could seem like a paradox as people need to be rewarded financially to pay for their basic needs. But the answer to this issue lies in thinking beyond our conventional notions of work and financial compensation. These people have been coding, for the most part, for years and do not see programming as bothersome. Raymond [1999] suggested that "*Every good work of software starts by scratching a developer's personal itch*", this shows that programming is also about empowerment and overcoming problems. Open-source projects start with a frustration, searching a solution to a problem; in fact, if you identify a problem, then you are usually well-placed to start solving it and find a better solution. If your itch is a common itch, and you know people that have a partial or complete solution to that problem, then you can likely build a community around that.

Shah [2006] noticed however that the community assumed that developers contributed to open-source projects either for financial incentives or need-based incentives. But he identified a third source of motivation: "*fun and enjoyment derived from the very act of creating and tinkering*". This category of hobbyist might create differences in the way a software is made; someone seeking for financial rewards may design a project to be of interest to a large market segment, whereas the hobbyists driven by enjoyment and challenge might seek to explore new possibilities and thereby creating innovations that are considered novel.

When a program is closed-source, the developers may not know the problems intimately the way users do. What started with Satoshi Nakamoto and its vision, compelled many developers and brought them into the heart of his solution model. It is also very unlikely that Bitcoin could have been released closed-source [Franco 2014]; otherwise, its creator could have easily inserted code that would have deviated from the original specifications. This would have allowed for instance the creation of new coins without authorization and thus, compromising the whole vision of decentralization.

According to Schneier [September 15, 1999], open-source is necessary in the cryptography world; In fact, proprietary security is usually less secure than public security. Because "*cryptography is hard to do right*", the only way to check if an algorithm or protocol works properly is by examining it. In cryptography, security usually has nothing to do with functionality as you can very well have two algorithms, one secure and the

other one not secure, and both working perfectly. These algorithms can encrypt and decrypt data without issue but the only way to truly distinguish their efficiency is by having them analyzed and tested by others. As a matter of fact, an algorithm can truly be considered secure after being examined by experts in that field over the course of years. One counter-argument mentioned in the paper is that "*secret cryptography is stronger because it is secret and public algorithm are riskier because they are public*". It indeed sounds logic, however public algorithms are by nature stronger because they are voluntarily made public; If a public algorithm works and is secure, then there are absolutely no risks in making them public in the first place. Consequently, if an algorithm is kept secret, it will only remain secure until someone reverse-engineers and publicly publishes the algorithm.

To illustrate this argument, Shneier uses the U.S. digital cellular companies and how they decided to create their own proprietary cryptography. These companies wanted to generate their own cryptography algorithms based on confidential specifications. And every time this specification would be made public and broken, they would make a new one. This is why the U.S. cellular industry is now considering the use of public cryptographic algorithms to replace their current ones. As previously mentioned, algorithms need to be tested, however publishing the code publicly may not automatically mean that people will examine it and check for security flaws. Being public is a good thing but not a guarantee of security. In fact, proprietary programs could have fewer security flaws than public ones, however, it often results in the public one being more secure over time as people will discover more security flaws than the proprietary code. As an illustration of this, we could compare the Windows operating software, owned and developed by Microsoft, with Linux, open-source and accessible by anyone.

The label "*Open-Source Software*" OSS was created after the "*general public license*" GPL legal paper got released, which assured the availability of the source code of any software published under that licence [Hertel, Niedner, and Herrmann 2003]. The OSS label, however, uses a slightly different ideological concept in which in order to be considered as such, the source code must be made publicly available and permit changes by any user according to Raymond [1999]. This label enhanced the acceptance of open-source development by software companies. Consequently, a number of companies such as Netscape and Sun Microsystems decided to publish the source code parts of their software projects to benefit user contributions. We have other successful examples of open-source software products such as the Apache project, the Free BSD operating software, the Mozilla web browser and the most popular one, the Linux operating system.

The Fork concept has been a topic of research for some time in open-source software [Nyman 2015]. A fork is usually a situation where the developer community disagrees on the direction of the project. This results in the split of the project where some part of the code base is used and is made backward-incompatible with current versions. In fact, the management of online collaboration is less about coordinating tasks but more about overcoming conflicts that may arise between collective strategy and individual actions. In his study, Viseur [2012] discovered that majority of the forks he studied were

motivated by a need for technical specialization and that these forks rarely followed the extinction of the original project. A research by Robles and Gonzalez-Barahona [2012] shows that from an in-depth analysis of 220 forks, most of them can be friendly or competitive, as some of them can be made from the same original community but under a different name. By using the same study, we notice that forks have become more frequent over the years, as shown in the Figure 2.3.

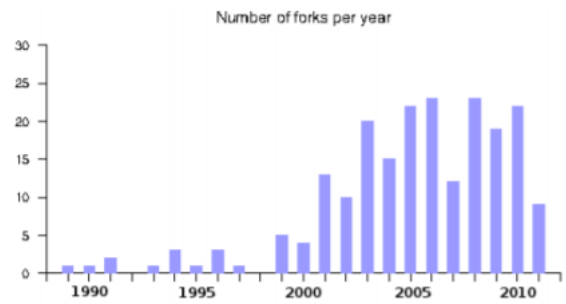


Figure 2.3: Number of forks created over the years [Robles and Gonzalez-Barahona 2012]

By reviewing the Chart 2.1, we can see the reasons these forks happen and can notice a category named -experimental forks- which might be seen as a subcategory of technical reasons. These types of forks occur when the developers of a community want to introduce major changes to the project, while others prefer a more conservative approach. In this situation, instead of just opening a branch that would be very difficult to merge in the main branch, a fork is created. These types of forks could be labeled as friendly forks, as many of them come with the approval of the original community. But in overall, the main reason behind a fork is for technical purposes.

Reasons	Frequency
Technical	27.3%
Discontinuation of the original project	20%
Community-driven development	13.2%
Legal issues	10.9%
Commercial strategy forks	9.1%
Differences among developer team	7.3%
Experimental	2.3%
Unknown	10%

Table 2.1: Classifying the main reason given for the forks

Our study will focus on several elements inspired by the aforementioned studies.

Most of these studies have focused on the different aspects of the Open-source Software community and their motivations and others on the economical aspect of Bitcoin.

Chapter 3

Research Design and Method

3.1 Data Sources

The main focus of this study is to understand and explain the development of the Bitcoin open-source software and identify a relevant relationship between its development and the price of the Bitcoin currency. The Bitcoin software was originally stored on SourceForge but switched to GitHub afterward to allow the community to contribute more easily to the project. To retrieve the historical data of Bitcoin in GitHub, we used two sources; GhTorrent and GitHub Archive. The official GitHub API has some constraints, most notably the maximum number of API requests per hour, which is currently fixed at 5000 for most purposes. As we need to experiment with large datasets, we only use this data source for specific experiments when the data is not available on both GhTorrent and GitHub Archive.

3.1.1 GhTorrent

GhTorrent is an off-line mirror of the data available on the official GitHub API [Gousios 2013]. The GhTorrent project has been collecting data since February 2012 and posts monthly dumps of data that anyone can use for research purposes. The data is available in an unprocessed format, in a MongoDB database and the metadata is extracted and stored inside a MySQL relational database. The GhTorrent data-set covers a broad range of development activities on GitHub but our focus will be put on the Bitcoin repository. The GitHub repository of Bitcoin was created December 19th of 2010 and the data collected from ghtorrent goes until March 2018, which gives us almost 8 years of data to analyze for this paper.

3.1.2 GitHub Archive

Created by Grigorik [2012], GitHub Archive is a project that records the public GitHub activity, archive it, and make it easily accessible for further analysis. GitHub Archive

polls data from the GitHub main feed as often as possible for information about recent events and captures only information about activities in the past few moments. This makes GitHub archive much richer in data, but on the other hand harder to work on as we the quantity of data to browse through is bigger. However, the data available on GitHub Archive concerning the Bitcoin project only starts in 2015, which is why we only use this source of data for specific experiments.

3.2 Retrieval Strategies

For the first research question, in order to produce an accurate description of the Bitcoin development process, we use the official *Contributing to Bitcoin Core* documentation, which has been written over time by contributors of the project. We have to note however that this document was not available since the genesis of the project, meaning that the workflow of Bitcoin may not have been as organized as today. For additional data, we will refer to developer notes and posts from the official Bitcoin forum bitcointalk.org where the community and developers communicate with each other.

For the rest of the study, we used the datasets available on Google BigQuery for both GhTorrent and GithubArchive, in order to retrieve data for our analysis. The data needed for each research question would be queried on specific tables with Legacy or Standard SQL queries depending on the complexity. When executing a query, BigQuery allows us to select a specific snapshot of the dataset; in this paper, for GhTorrent we use the snapshot of April 2018 and for GitHub Archive, we use the yearly snapshot of 2015, 2016 and 2017. In fact, a snapshot of GhTorrent will store the data of every GitHub event up until that point, whereas GitHubArchive will only store data for a specific time-frame (that could be daily, weekly, monthly or yearly).

We have to note however that GhTorrent may have a slight marge of error due to malfunctions in the GhTorrent mirroring system (software or network) that can result in some missing parts of the data. Similarly, the GithubArchive may have some data missing due to its data collection process that only captures data from the last few moments; it needs to constantly poll GitHub or risk losing data (and this may still happen if there are unexpected spikes in activity on GitHub). These defects may create a slight margin of error during our experiments.

What's more, GhTorrent and GitHub Archive are fairly different in the way they store data; GhTorrent splits the GitHub data across multiple tables with a mirroring algorithm, based on a recursive dependency resolution process, whereas GitHub Archive will store JSON encoded events as reported by the GitHub API inside a single table. Because of this, we will prioritize the use of GhTorrent as it allows us to select and filter data on specific tables, without having to fetch the entirety of the dataset.

For research question 3, we could not retrieve the exact number of opened/ closed/ merged pull-requests on GhTorrent due to the previously discussed malfunctions. The data was instead retrieved by using a Python script on the official GitHub API, analyzing each pull-request made in the Bitcoin project individually and its state over time. We then do a similar experimentation with GitHub Archive, which condenses the GitHub

API data into a single SQL table, and we will use it mainly for sampling and sample Bitcoin pull-requests in the year 2015, 2016 and 2017.

Bitcoin can be viewed as both a software and a digital currency. For retrieving the economical data of Bitcoin, we wrote Python scripts to extract the daily average USD market price of Bitcoin across major Bitcoin exchanges from the data source of Blockchain.info. Manual inspection was used to resolve such things as multiple email addresses in cases where the automated technique failed. The reason we use this website as our source is due to the fact that it updates its Bitcoin price database based on the average daily price of Bitcoin on every financial market it is available in. Afterward, we computed the average price for each month in order to have a lighter set of data to analyze. Thanks to this average calculation, we decrease our data-set from 2753 values to 92 values.

For the experiments of research question 4 and 5 that review the economical aspect of Bitcoin, we focus the experiments on the Bitcoin price data and we compare it with the results of previous experiments we made from the analysis of Bitcoin development.

Chapter 4

Results

4.1 The Bitcoin development process

RQ1: *What is the process used to develop Bitcoin?*

The Bitcoin software was originally released in 2009 on Sourceforge by Satoshi Nakamoto as the first proof-of-concept of peer-to-peer electronic cash. It became shortly available on GitHub where its development continued. The Bitcoin software development project operates an open contributor model where anyone is welcomed to contribute in the form of peer review, testing, and patching.

Even though the Bitcoin-core project is an open-source project, where anyone is free to contribute, some hierarchy seemed necessary; instead of using the term of "*core developers*" which seemed to give privilege to people, the role of "*maintainers*" is used. These "*maintainers*" are responsible for merging pull-requests and are moderated by a "*lead maintainer*" who is responsible for the release cycle, overall merging, and appointment of maintainers.The role of "*maintainers*" used to be given out to people contributing frequently to the Bitcoin repository and revoked once they became inactive. However, it seems to be slightly different nowadays, where a frequent contributor will get attributed the role of "*maintainer*" of a specific set of functionality and will get the commit access restricted to this set only. Two roles were then created; "*GUI maintainer*" in November of 2015 and "*QA/Testing Maintainer*" in April 2016 to increase the efficiency of the maintaining schedule process in specific functionalities.

The original creator of Bitcoin, Satoshi Nakamoto, did not use a Version Control System at the beginning; the releases and source code were released inside a RAR file and uploaded to bitcoin.org. A community member then helped Satoshi Nakamoto to set up the original repository on Sourceforge for him and later on GitHub. However, Satoshi still had control over the original code and every patch authored by community developers were emailed to Satoshi himself and the changes were then committed manually to the source with the commit message containing the attribution but not the actual commit itself. This is after the introduction of maintainer positions that this process got streamlined on GitHub.

The Bitcoin project follows the pull-based software development model; for every developer who wishes to contribute to the development of Bitcoin, they can do so by using pull requests which facilitate the local development, testing and peer review. Pull-requests form a new method for collaborating on distributed software development in which the contributor forks (or clone) the project's repository and make their changes independently from the main project. The code is then reviewed by the community and must be approved by one of the "maintainers" of the project in order to merge the changes. Community code reviews are a core aspect of the pull request development model. One important thing, however, is that pull requests made by contributors must be focused on a single task; this being a feature, fixing a bug or refactor code. A mixture would make it overly complex and difficult to review by the community. Pull-requests form a new method for collaborating on distributed software development in which the code reviews are a core aspect of the pull request model.

The developer community is free to work on any aspect of the Bitcoin software they deemed important or necessary. However, the Bitcoin developer mailing list is used to discuss complicated or controversial changes before working on a specific issue. Since anyone can subscribe to the mailing list, the propositions and conversations are reviewed by many people from and outside the core development community. Furthermore, the developer has to keep in mind that when adding a new feature, he has to think in the long-term technical maintenance of it; they have to consider if they are willing to maintain it on their own, find bugs, and manage it to avoid its feature being orphaned by the community. In fact, if a feature is no longer maintained in the project, it gets removed by one of the appointed *maintainer*.

Finally, comes the "*Decision making*" process; this process defines whether a pull request is merged to the Bitcoin Core project or not. Maintainers of the Bitcoin core project will review each pull request and verify if it meets the minimum standard required for inclusion, those being:

- As talked earlier, the pull request must have a clear use case, that either fixes a bug or adds a feature that can be maintained in the future.
- Have implemented unit and functional tests where necessary.
- It must follow the code style guideline defined by the *maintainers*.
- It must not break or negatively interfere with the current code-base.
- The pull request must be well peer reviewed and be approved by the community.

For the peer review process, anyone is allowed to participate and may do so by commenting directly inside the pull request. Reviewers have to pay attention to code errors and test the patch. Project maintainers will then consider every peer review when determining if a pull request should be merged to the main branch. A specific code language has been established to easily identify important feedback and can be seen in Table 4.1.

Code	Meaning
ACK	"*I have tested the code and I agree it should be merged*"
NACK	"*I disagree this should be merged", and must be accompanied by sound technical justification (or in certain cases of copyright/patent/licensing issues, legal justification)*"
utACK	"*I have not tested the code, but I have reviewed it and it looks OK, I agree it can be merged*"
Concept ACK	"*I agree in the general principle of this pull request*"
Nit	"*trivial, often non-blocking issues*"

Table 4.1: Peer Review vocabulary table from the official *Contributing to Bitcoin Core* documentation

Through this vocabulary, we can analyze later on this paper the behavior of the peer reviewers and the statistics in regards to the pull requests.

When a developer proposes a patch to change the consensus of Bitcoin, some additional steps are necessary; first off, the developer must write a BIP (Bitcoin Improvement Proposals). Similar to Python's PEP (Python Enhancement Proposals), BIP (Bitcoin Improvement Proposals) are a design document providing information to the Bitcoin community or describing a new feature for Bitcoin itself. This Request For Comments (RFC) process is the primary mechanism for proposing new features, collecting community input on an issue, and for documenting the design decisions that go into Bitcoin. Once the developer discussed this issue extensively with his peers through the IRC and the Bitcoin core mailing list in order to ascertain whether the idea is BIP-able, he can create a pull-request to work on it. Additionally, updating Bitcoin to a new consensus requires every people on the network to run that new code, otherwise, the whole network won't get properly updated and it may cause issues. The Bitcoin Core community has generally been very conservative and cautious when it comes to these sorts of changes. These changes not only require a deep understanding of the Bitcoin protocol but also the majority of the community to validate and follow this new consensus. This is why consensus changes that are likely to cause divergent interests are almost always avoided by the community despite being a recurrent topic.

Through this development process analysis, we can see many similarities between the open-source development process of Bitcoin and other open-source projects. Bitcoin uses a mailing list and an IRC channel similarly to the Python and Apache open-source projects, where developers can communicate and discuss new improvements

and fixing bugs. What's more, the Bitcoin project uses a similar approach to the FreeBSD's project management; when a developer consistently contributes to the projects, he may be qualified to become a core member of the project. In the case of Bitcoin, contributors can become maintainers in a specific set of functionality, however, the result is the same as these members get specific privileges such as being able to approve the addition of new features, as well as being able to reject them if inappropriate. Furthermore, despite Bitcoin being a community project, where every new addition or modification to the project need to be approved by the community, the Bitcoin maintainers still have the overall control and are the ones making the final decision. In addition, the Bitcoin project uses the same Pull-based development process as the Python project: every user interested to work and contribute to the open-source project has to create a duplicate of the original repository where they can isolate their modifications and work in their own environment. When a set of changes is ready to be submitted to the main repository, they individually have to create a pull request, which specifies a local branch to be merged with a branch of the main repository. A member of the core-project (in the case of Bitcoin, the maintainers) then has to manually inspect and review the changes; if he approves them, they are merged into the main branch, otherwise, the contributor has to revise it. Finally, for more advanced feature implementations, the Bitcoin project uses BIP (Bitcoin Improvement Proposals) which are documents written by community members that propose new features, similar to Python's PEP (Python Enhancement Proposals).

In the next section, we give a closer look at the distribution of development, the community involvement, and the impact of the economical value of the Bitcoin currency in the developer community.

4.2 The team behind Bitcoin

In this section, we present the different results from our GhTorrent data analyses focused specifically on the Bitcoin project.

RQ2: How many people are working on the Bitcoin software?

The philosophy of the Bitcoin software is decentralization, and it is interesting to see if its development follows the same mindset. The Figure 4.1 displays on a world map the percentage of user who participated at least once in the project.

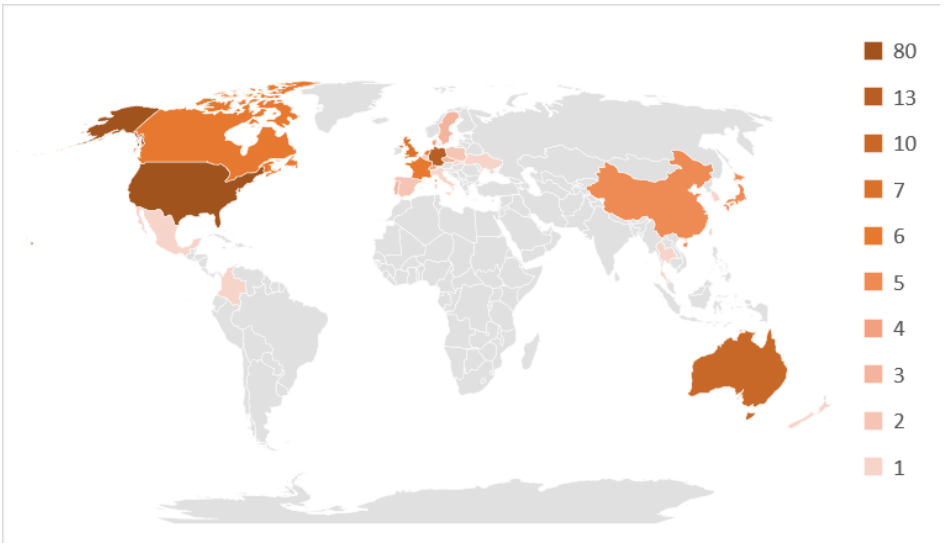


Figure 4.1: Mapping of Bitcoin developers by country

The Figure 4.1 shows that the developers are highly concentrated on specific part of the world. Those mainly being the United-States, Germany, and Australia. Other parts of the world such as China, Europe in general and Canada still see some activity. The United-States still occupies a large part of the Bitcoin development community as over 44% of the global developer community is working from the United-States. One issue with this statistic, however, is that only an average of 9% of all the registered and verified GitHub accounts have set their location on their profile. Taking this issue into account, we gather the total number of GitHub accounts who have committed at least once on the Bitcoin project and we find a total of 442 accounts.

Based on our previous and figure 4.1, 183 GitHub accounts who participated in the Bitcoin project have set their location. This gives us an estimated 43% of developers showing details of their identity, way above the global 9% of average users. It also indicates that these users are not afraid to show parts of their real-life identity, on such a tremendous project with real financial consequences.

Next, we look at the number of commits made by country. We previously analyzed the global localization of the Bitcoin developer community, but we need to know which country is contributing the most on the project; Figure 4.2 maps the degree of commit activity by country, similarly to Figure 4.1.

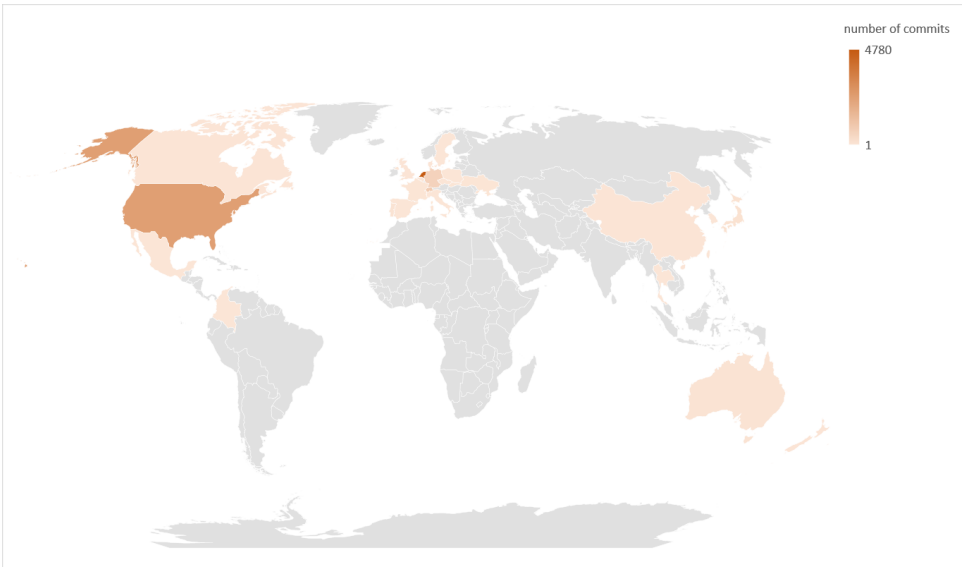


Figure 4.2: Mapping of Bitcoin commits by country

Figure 4.2 shows that the number of commits is highly concentrated in the Netherlands, which did not figure in the top 3 most active Bitcoin developer country. With a global total of 9,333 commits among all of the GitHub accounts who have registered their location, Netherlands counts a total of 4,780 commits which represents around 51% of the global commits. This means that half of the commits made of Bitcoin are made from a single country.

Given the regularity of Bitcoin commit activity over-time, we want to investigate how the community actually participates in the development of the Bitcoin-core software. In second position we see the United-States, with over 2,414 commits and in third position Switzerland with 889 commits.

By taking both collected results from before with Figure 4.1and Figure 4.2, we create Figure 4.3 which allows us to better visualize the contrast between the number of users participating actively in the Bitcoin development and the number of commits made by country.

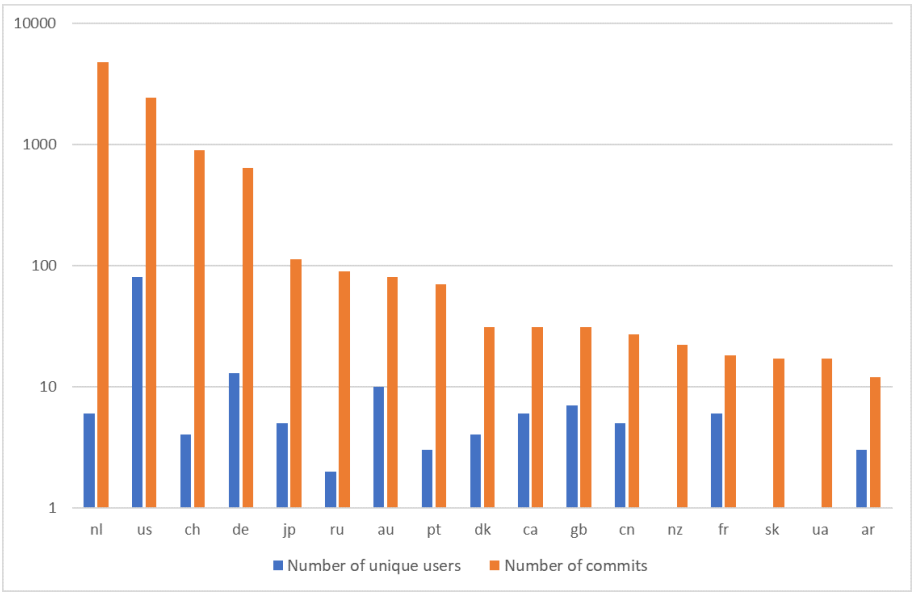


Figure 4.3: Number of users compared to the number of commits by country

The data (Figure 4.3) confirms our previous observation as a high number of users does not necessarily imply a high number of commits. The United-States is the only exception; Although it has the highest number of GitHub users, the average number of commits per user is around 31 whereas the Netherlands is around 797 and Switzerland is around 223 commits per user. Other countries such as Australia and Great-Britain, show an above average number of users working in the Bitcoin project. They, however, do not seem to have an important part in the global development of the project. We can confirm through these observations that despite the global interest of Bitcoin, only a few numbers of developers from all around the world are actually contributing to the project.

By the results we get from Figure 4.3, we can speculate that the core team is working in the top 3 contributor countries, with the addition of anonymous developers. Figure 4.4 shows the list of developers who most contributed to the Bitcoin core project since its inception on GitHub.

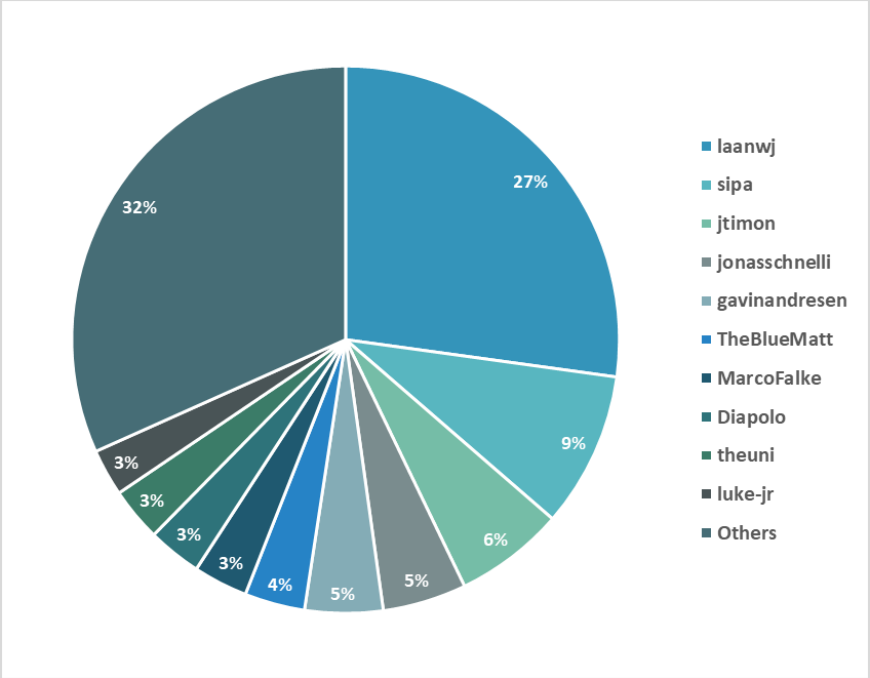


Figure 4.4: Bitcoin most active contributors

Wladimir, or by his GitHub pseudonym laanwj, is the most active developer on Bitcoin by accumulating almost a third of all the commits made in the project. What's more, the top 10 developers of the project account for 68% of the total number of commits made in the Bitcoin core project. This number reveals that a very small number of developer are responsible and actively contributing to the project. As we can see, a single core developer is contributing more than 30 % of the Bitcoin project by himself, we should review over the years the participation of the overall core members of the projects.

We retrieve a total of 442 unique developers that participated in the Bitcoin project since its creation, however, not all of them have contributed equally to the project.

Figure 4.5 shows the cumulative distribution of the most active Bitcoin developers based on the number of commits they individually made throughout the Bitcoin history. We selected for this distribution, all the developers who had at least contributed once in the Bitcoin project. In total, we have 442 unique developers that participated in this project and that we can analyze.

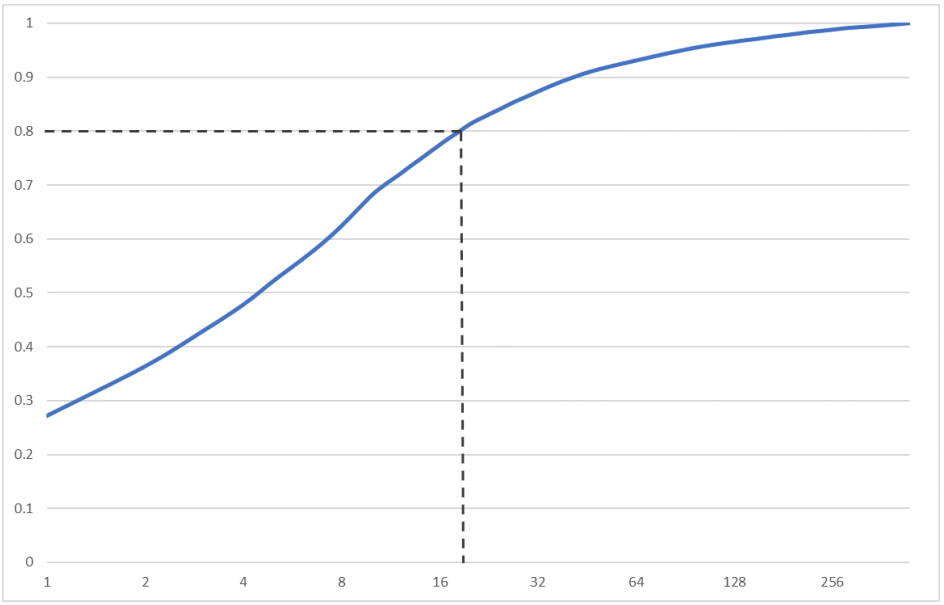


Figure 4.5: Cumulative distribution of active Bitcoin contributors

Figure 4.5 indicates that 18 developers contributed around 80% of the total number of contributions. For simplicity and based on the hypothesis of Mockus, Herbsleb, and Fielding [2000] in which: "*Open source developments will have a core of developers who control the code base. This core will be no larger than 10-15 people, and will create approximately 80% or more of the new functionality*", we will call these 18 contributors core-developers as they are the most active ones in this project. What's more, we can observe that passed the 80% of contribution, the number of contribution becomes smaller and highly concentrated. This indicates that from the non-core developers, only a small number of contribution are made by them, which means very small-effort is globally made by them to the Bitcoin project.

Based on the observations made on Figure 4.5, we concluded that the Bitcoin core-developer team was composed of around 18 members. Figure 4.6 shows the 18 most active contributors to the Bitcoin project year-by-year. The vertical axis displays the year and the horizontal axis shows the cumulated number of commits made by the 18 most active developers of the targeted year. The Figure 4.5 also shows the series connection lines of every developer contributing in the following years. This Figure gives us a better idea of the global amount of contributions made by the core-team but also the regularity of the contributors working on the Bitcoin project.

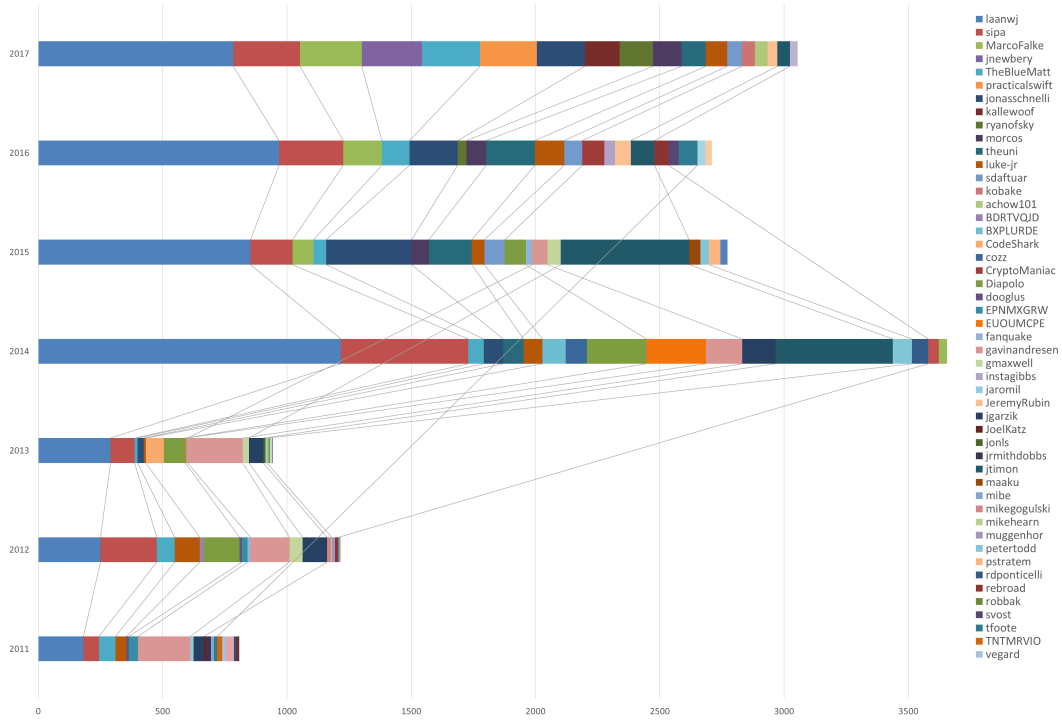


Figure 4.6: Bitcoin core-team participation by year

Despite only selecting the 18 most active contributors each year, we have a total of 62 developers who were, at least once, considered in the core-team of the Bitcoin project based on our 80% contribution activity assumption. With the help of the connection lines, we can compute the attractiveness of the project by measuring the percentage of developers continuing to contribute to the Bitcoin project from one year to another. In seven years, we have an average of 56% of the contributors still participating actively in the core team. This means that each year, around 44% of the developers drop from the core-team and either stop completely to contribute or do not make enough significant contributions.

Furthermore, we can observe that since the genesis of the project, only 3 developers have been constantly present in the core-team until now, and 37 developers have been in the top 18 core-team only once. Out of the total 62 contributors we collected, 25 of them have been more than once in the core-developer team of Bitcoin, and only 3 of these contributors have been working on the project at a regular pace since the beginning.

Between 2011 and 2013, we can observe an average of 970 contributions per year made by the core team, with a percentage variation of around 25%. However, we can notice a sudden large growth of contributions in the year 2014, with a contribution increase of 291.94% compared to 2013 which corresponds to an increase of around 2,715 contributions in one year. Afterward, between 2015 and 2017, we note a yearly average

of around 2,720 contributions made by the core developers.

We previously assessed that they were different roles attributed to frequent contributors of the Bitcoin project. From 2010 to 2016, the Lead Maintainer of the project was Gavin Andresen (gavinandresen). This role was not attributed but more so grandfathered by Satoshi Nakamoto himself after his disappearance. Gavin was indeed the most active contributor of Bitcoin at the beginning of the project until 2014. Gavin Andresen was also the community member who originally helped Satoshi Nakamoto to set up the original GitHub repository of Bitcoin.

After further inspection on the history of the repository, the lead maintainer access of Gavin got removed from his account in 2014 due to his desire to focus on Bitcoin research instead of continuing its development. The following Lead Maintainer of the Bitcoin project is Wladimir J. van der Laan (*laanwj*) who since 2011, has always been a very active and contributing member of the project. According to Martin and Yehuda [2016]: "*the only way to gain leadership on an Open-source project is to earn the role within the community. And the only way to do that is to gain credibility and make contributions*". Over the years, Wladimir J. van der Laan proved his capacity to contribute regularly on the project.

From the official presentation paper of the GhTorrent project, the leader of a project is identified as "*the individual with 20% more project contributions than the one with second highest*" [Grigorik 2012]. This statement is not true for the year 2012 and 2013 where the Lead Maintainer was below the number of contribution from Wladimir J. van der Laan (laanw) who then replaced him afterward. The 20% is, however, true after the change of Lead Maintainer in the project as Wladimir J. van der Laan (laanw) holds an average of 170% more project contributions than the one with the second highest since then.

Jonas Schnelli (*jonasschnelli*) became the the GUI Maintainer on 2015 from his large participation during the year of 2015 that can be noticed on the Figure 4.6. Marco Falke (*marcofalke*) became the QA/Testing Maintainer on 2016 and his increase of activity is noticeable as he became the third most active developer the following year, in 2017. The project still keeps on the other hand the 2 main core developer of the previous year. Ultimately, the organization of the project among its developers got changed over the years up until now, with a clear leader and maintainers, working actively on the project, each with a clear focus on their set of functionalities.

Another way to identify the core-developers of the Bitcoin project is by analyzing and retrieving the GitHub accounts linked to the official Bitcoin account and repository as project members. Figure 4.7 retrieves the members that were added to the GitHub Bitcoin organization. Being a member of a GitHub organization implies this member has special privileges such as being able to directly push commits or merge and close pull-requests.

As we saw in the Bitcoin development process section, these permissions are given to frequent contributors who become maintainers of the Bitcoin project. What's more, even though this organization is listed as private, we know that there are multiple maintainer teams, each focused on a particular feature. However, this restricts us on

the amount of data we can actually retrieve as only a very few snapshots of the Bitcoin organization page were made.

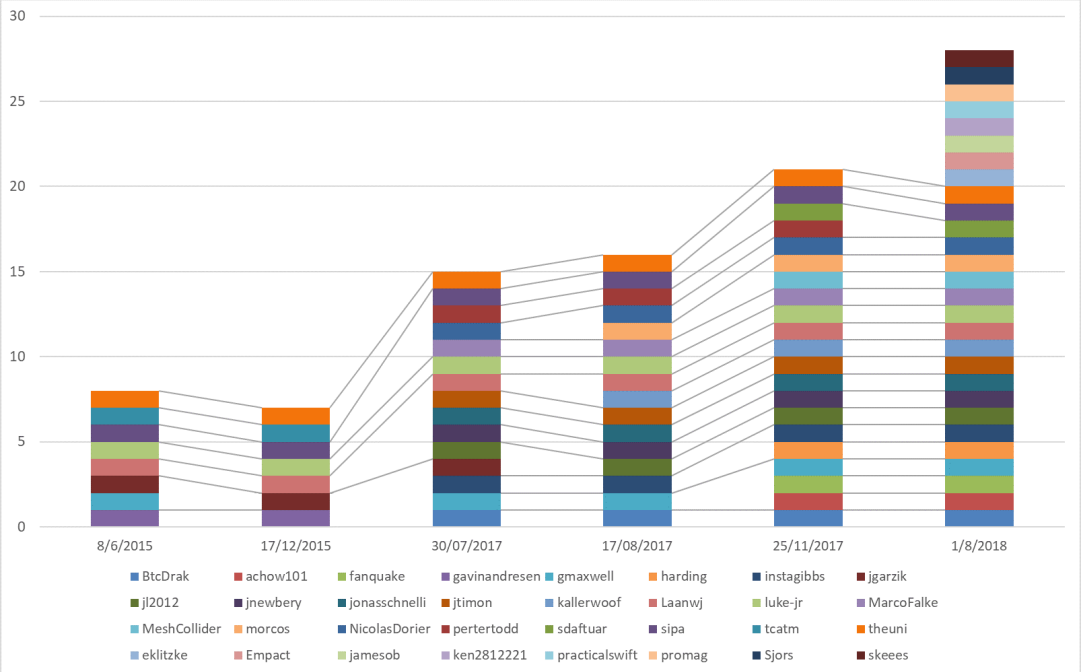


Figure 4.7: Bitcoin organization members at multiple point of time

As discussed earlier, only some instances of the year 2015, 2017 and 2018 can be seen in the Figure 4.7. However, we can identify a trend in term of organization size; in fact, between 2015 and 2017, the organization of Bitcoin doubled, growing from 7 members in December of 2015 to 15 members in July of 2017. Furthermore, up until August of 2018, the organization members stay the almost the same, and despite having a member or two leaving his responsibility from the organization, the organization community stays faithful and steady. In addition, we can see many newcomers to the Bitcoin organization in 2018, growing from 21 members in November 2017 to 28 members a few months later in August 2018.

In conclusion, we were able to identify with Figure 4.4 and Figure 4.5 that the development of Bitcoin was indeed concentrated within a small number of developers. These core-developers are located within particular geographic locations as discussed on Figure 4.2, mainly in the Netherlands, United-States, and Switzerland. Over time, the core-developer team of Bitcoin spilled, with contributors leaving and new developers replacing them. However, we can see that the Bitcoin organization keeps growing, by adding more new members each year as maintainers, allowing the team to expand.

4.3 Measuring the Bitcoin project activity

RQ3: *Is the Bitcoin open-source software an active community project?*

In this section, we will have a look at three aspects of the Bitcoin project in order to measure the project activity:

- The number of contributions made in the project.
- The community activity and their interactions.
- The pull-requests merge rate.

4.3.1 The Bitcoin contributions

A first good measurement to evaluate if an open-source project is active or not is by analyzing the number of contributions made by the community. In fact, if nobody contributes to an open-source project, then it won't be able to attract new developers.

Figure 4.8 shows the number of Bitcoin commit activity by quarters since its creation.

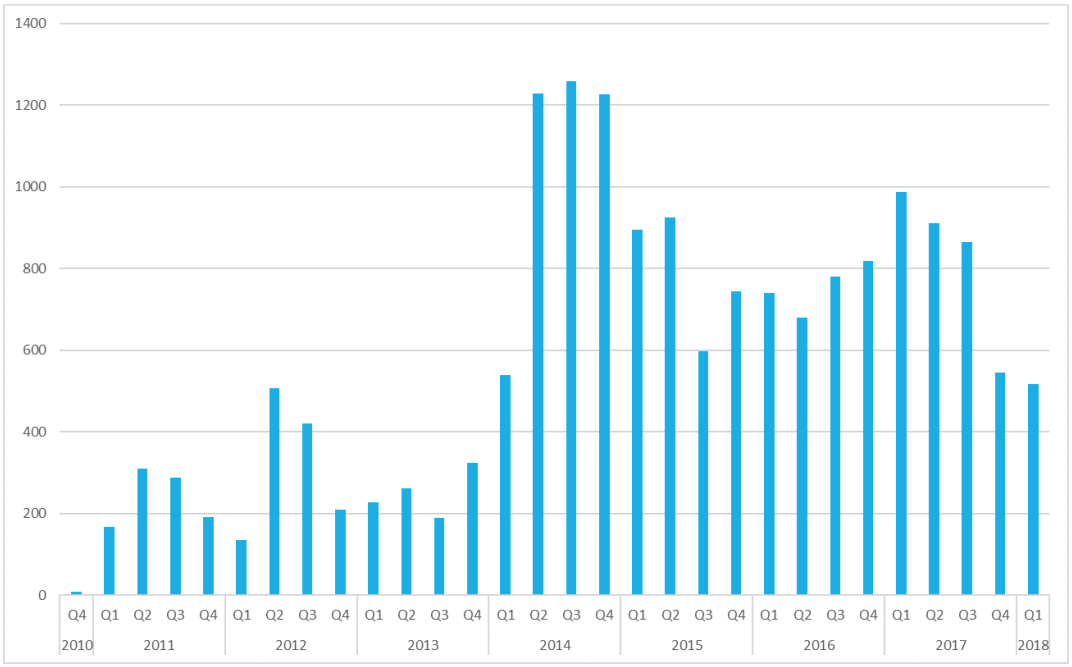


Figure 4.8: Bitcoin commit activity by quarters

We can observe that the commit activity is moving in three major peak waves; the first one in Quarter 2 of 2012, with a high of 507 commits in that period. It then stabilizes at around 222 commits per quarter until 2014, when it reaches over 1,260

commits for 3 quarters. It is an increase of over 467.57% commits from 2013 to 2014. The number of commits then hovers at around 700 commits per quarter until moving up once again in the first quarter of 2017, reaching 988 commits. After this increase of 38% of contributions, the number of commits slows down the following quarters.

We previously discussed how concentrated the development of Bitcoin was, with an average of 18 core-developers working intensively on the project. This, however, does not limit a steady increase in the number of commits and contributions to the project.

Figure 4.9 displays the number of unique users making commits each month to the Bitcoin project and contributing to its development.

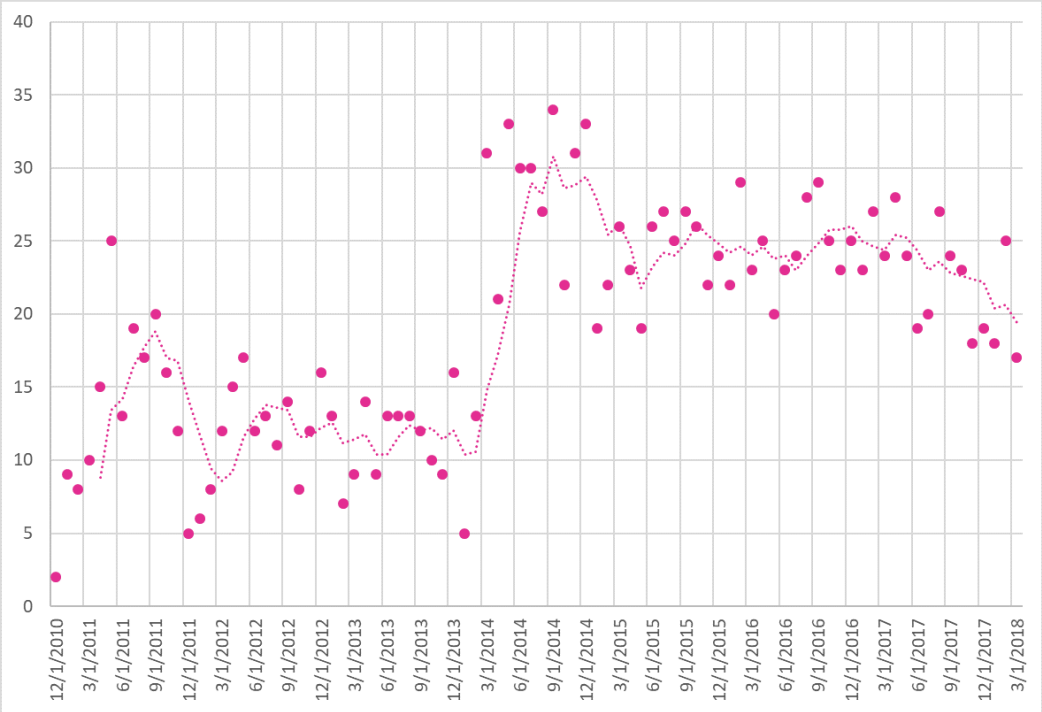


Figure 4.9: Number of unique users making commits to the Bitcoin project for each month

In the case of Bitcoin, we can see in the Figure 4.9 a slow start, with an average of 13 unique users contributing each month by making commits. It is until 2014 that we see a sudden increase in users participating in the project, reaching an all-time high of 34 unique users making commits in a single month. However, this number slowly started to drop since then, describing a current downtrend of active users participating on the project. Despite the decrease being low, this may indicate that the Bitcoin project becomes more and more exclusive, with the core team contributing more to the project than the rest of the global contributors.

By looking into the number of commits from Figure 4.8, and by taking into con-

sideration the results from Figure 4.9, we see a very similar trend, as the number of commits decreases, the number of unique contributors decreases as well. This may indicate that the developers contributing to the project are usually the same and that fewer contributors are joining the project over time, however, those that join it stay in for the long haul. Adding the fact that on Figure 4.7, the Bitcoin organization adds more and more members as maintainers to the project, despite the decline of contributions, the project still introduces new talents on the development of the Bitcoin project.

4.3.2 The Bitcoin community activity

Most papers analyze the number of contributions made by developers instead of focusing on other important aspects of an open-source project. This is why we will analyze the community activity inside the comments posted in the pull-requests. In fact, the Bitcoin project has adopted a pull-based development in which contributors can work in their own cloned distributed environment and submit it to the core team that oversees the project to review it, provide feedback and merges it if the contribution is accepted.

Figure 4.10 shows the total number of comments posted inside pull-request threads every month in the Bitcoin project on GitHub. These comments are a good indicator of activity as they reflect the global participation of the community in discussing and reviewing new features or improvements inside the pull-requests. We added a trend-line in the graph calculated with the least squares to visualize the increasing steady rate.

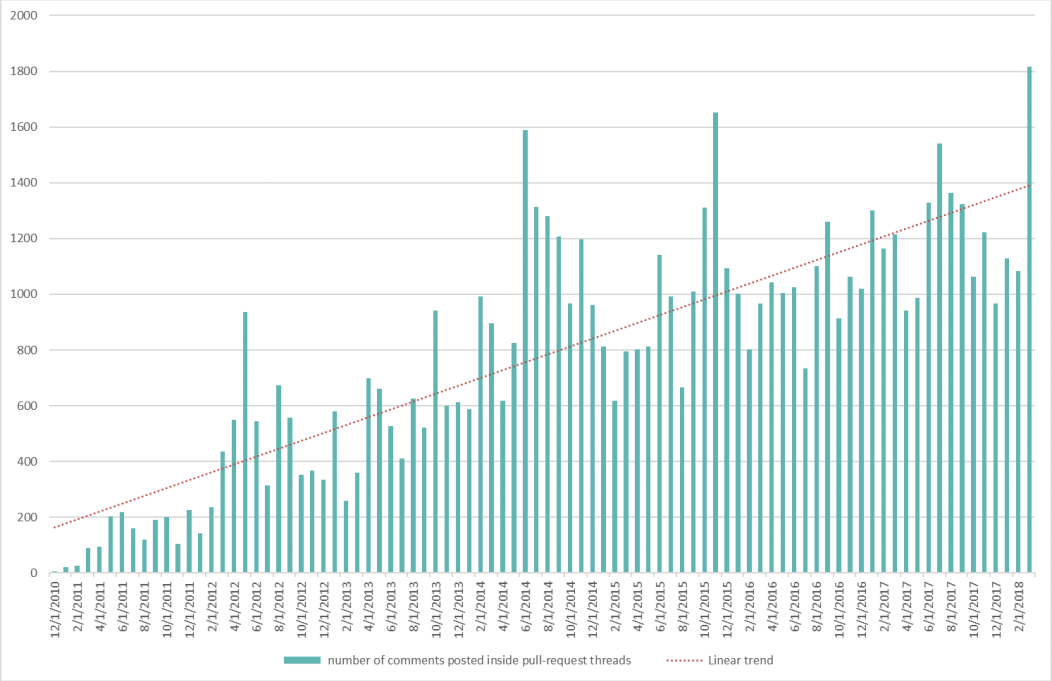


Figure 4.10: Number of monthly comments posted inside pull-request threads with linear trend

The number of comments posted by the community naturally grows over time, confirmed by the linear trend-line. Despite a drop of comments on some months (notably at the beginning of the year 2015), the global trend stays strong and keeps increasing over time.

To put that previous result in perspective, Figure 4.11 shows the total number of unique members commenting inside Bitcoin pull-requests each month. As we did for Figure 4.10, we added a linear trend calculated with the least squares to show in which direction it evolves over time.

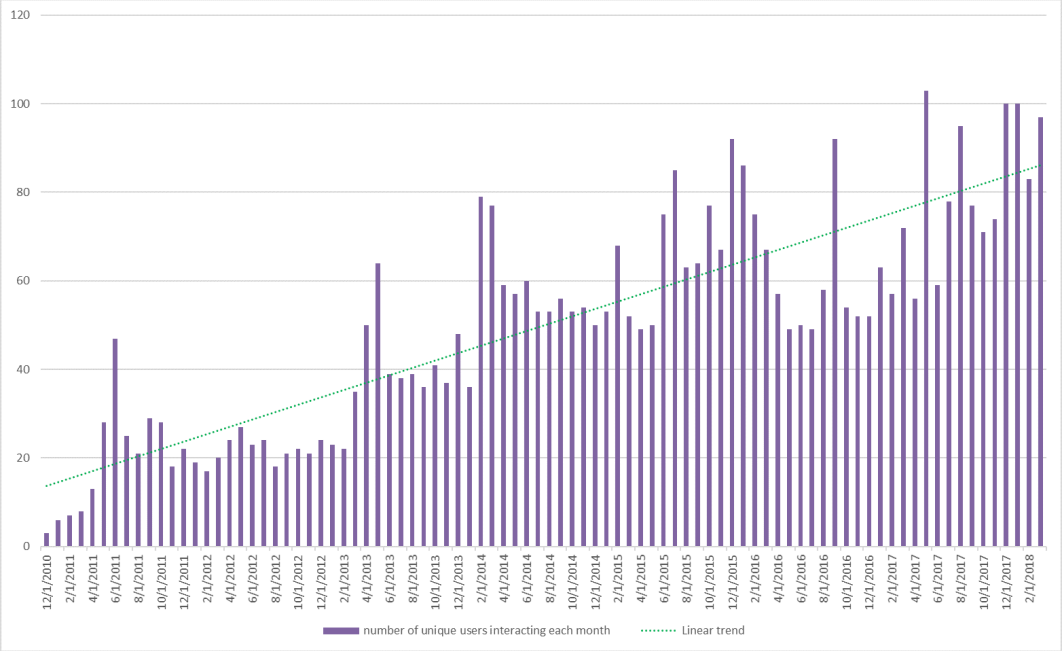


Figure 4.11: Number of unique users interacting each month with linear trend

The trend of Figure 4.11 is similar to the trend of Figure 4.10. To confirm this observation, we create Figure 4.12 which blends the previous figures 4.10 and 4.11 and adds a linear trend computed from the ratio of comments per unique user.

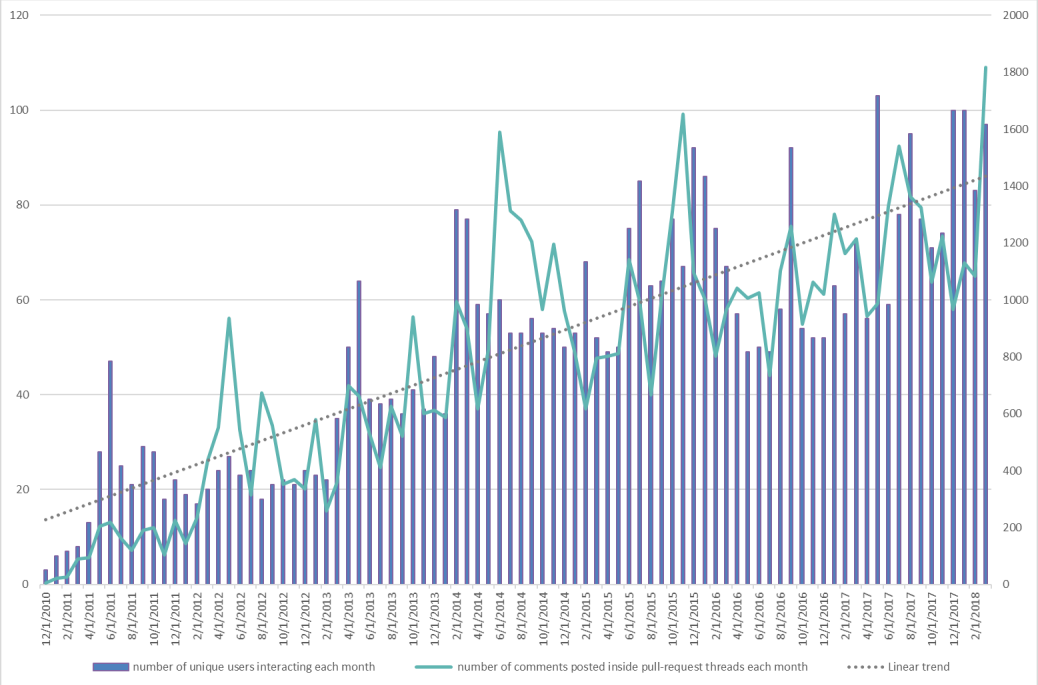


Figure 4.12: Blending the number of unique monthly users with the number of thread comments posted inside pull-requests and showing the global trend

Despite the increase over time of active users interacting with each other in the Bitcoin community seen on Figure 4.11 and the increase of comments posted on Figure 4.10, the average number of comments posted by user is still hovering in the same range, between 9 and 27 comments per user. In fact, even though the trend of active users and the trend of number of posted comments are both increasing at a steady pace, they are both compensating each other (as we can see with the trend-line crossing their average) which indicates that the community is not limited to a small number of users posting a large number of comments by themselves, but also that the users are communicating at a decent pace with each other.

4.3.3 The Pull-based Bitcoin development analysis

In this section, we will analyze data from the pull-requests made in the Bitcoin repository.

The Figure 4.13 plots for each year the number of opened, merged and rejected pull-requests since the beginning of the Bitcoin project. These pull-requests are created and opened by the community and are merged into the original branch by maintainers of the project once they are approved by the community.

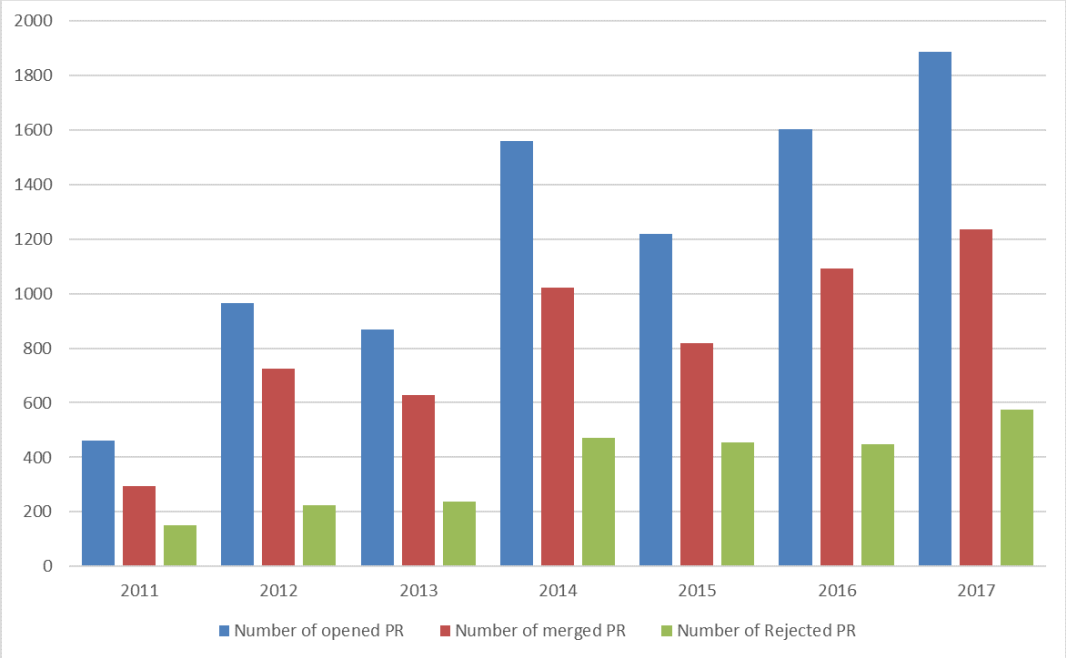


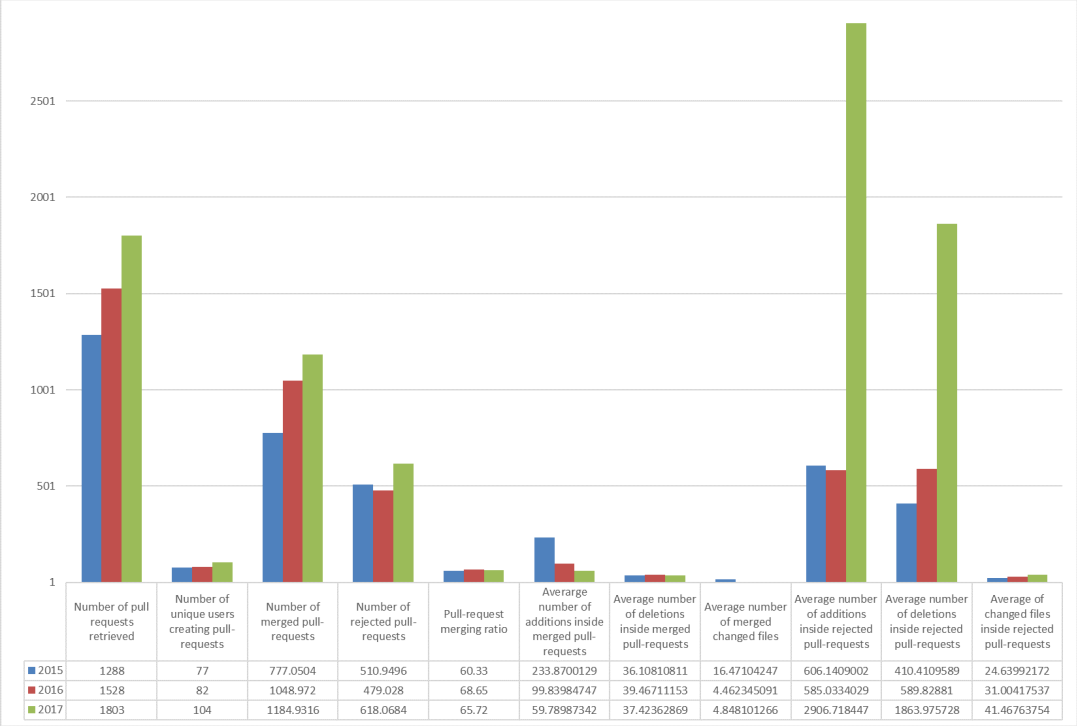
Figure 4.13: Comparing the opened/merged/rejected pull-request quota activity by year

The number of opened pull-requests grows steadily over time, from 461 in 2011 to 1,887 pull-requests in 2017. We can notice however a small decrease of 21% in 2015, with 1,219 opened pull-requests in total before going up again in 2016. From 2011 to 2017, the number of opened pull-requests increased by 309%.

The number of merged pull-requests follows very closely the trend of the total number of opened pull-requests, having the exact same variations at the same periods. Looking at the total number of commits over the years in Figure 4.8, we can observe a slight drop in the same period. The average percentage of merged pull-requests from 2011 to 2017 is around 68%, the lowest being 63% and the highest 75%.

The number of rejected pull-requests, on the other hand, has two range levels; the first one between 2011 and 2013 at an average of 200 unapproved pull-requests and the second one between 2014 to 2017 at around 487 rejected pull-requests per year. This second range level of rejected pull-requests was reached in 2014 and barely moved in 4 years.

Figure 4.14 gives a more in-depth analysis of the pull-requests activity by sampling a major number of pull-requests made in 2015, 2016 and 2017 and review multiple parameters in each of these years. Mainly, the number of unique users creating pull-requests, the number of merged pull-requests and finally the additions/deletions from both the merged and rejected pull-requests.



	Number of pull requests retrieved	Number of unique users creating pull-requests	Number of merged pull-requests	Number of rejected pull-requests	Pull-request merging ratio	Averarge number of additions inside merged pull-requests	Average number of deletions inside merged pull-requests	Average number of merged changed files	Average number of additions inside rejected pull-requests	Average number of deletions inside rejected pull-requests	Average of changed files inside rejected pull-requests
2015	1288	77	777.0504	510.9496	60.33	233.8700129	36.10810811	16.47104247	606.1409002	410.4109589	24.63992172
2016	1528	82	1048.972	479.028	68.65	99.83984747	39.46711153	4.462345091	585.0334029	589.82881	31.00417537
2017	1803	104	1184.9316	618.0684	65.72	59.78987342	37.42362869	4.848101266	2906.718447	1863.975728	41.46763754

Figure 4.14: Analysis of the pull requests activity with the sampling of the year 2015, 2016 and 2017

Between 2015 and 2016, we have an increase of opened pull-requests (which we also discussed in Figure 4.13) which consequently increases the number of unique users opening pull-requests by 35%. We previously deduced the average number of merged pull-requests on Figure 4.13 at around 68% and in this sample, we hover at an average of 65%.

Looking at the merged pull-requests statistics, we can see that the average number of additions made inside these pull-requests drastically decreases, with a drop of additions of over 75% between 2015 and 2017. Similarly, the number of changed files inside merged pull-requests also heavily dropped by 75%. However, the average number of deletions inside pull-requests stays constant, at around 37% in our sample analysis.

On the other hand, by looking into the rejected pull-requests, we can observe a strange phenomenon in the year 2017; while we do not see a huge evolution between the year 2015 and 2016 for the average number of additions and deletions for rejected pull-requests, we see however a huge increase in 2017 for both of these data. For the number of additions, we see an increase of almost 400% and as for the deletions, we see an increase of around 216%. The average number of changed files inside rejected pull-requests, however, does not follow this trend, with a small increase of around 8 changed files each year.

We can conclude through this sample is that over the years, despite the growth of

merged pull-request number, the number of additions made to the project suddenly decreases. What's more, in 2017, despite the increase of opened pull-requests, most of those that get rejected hold inside of them a large number of both additions and deletions which may indicate they contain drastic changes to the Bitcoin codebase.

Each update or change to the code has to be made with pull-requests, and before they get merged to the main code, they have to be approved by the community as explained in our first section. As we discussed previously with Table 4.1, the community have 5 keywords to review a pull-requests in order to determine if it should be merged or not. Table 4.2 displays the total number of keywords found inside pull-requests comments between the years 2015 and 2017.

Year	ACK	NACK	utACK	Concept ACK	nit	Total number of reviews	Total number of comments
2015	1,411	301	982	546	384	3,624	11,708
2016	1,580	298	1,866	780	408	4,932	11,934
2017	1,316	267	2,560	755	549	5,447	14,414

Table 4.2: Retrieving the total number of comment reviews made from 2015 to 2017 for each *Bitcoin Peer-review* keyword

Out of all the comments posted each year inside the pull-requests, we have an average of 36% of comments that are pull-request reviews with a keyword. Furthermore, the total number of comments increases as well as the number of reviews.

In 2015, the most used keyword was the ACK, with a ratio of 38% out of all the reviews posted that year. This indicates that more than a third of the pull-requests reviewed were thoroughly tested by the community. The utACK represented 27% of the total reviews made by the community. However, starting in 2016, we can see an inversion of ACK/utACK ratio of reviews meaning that more people reviewed and approved a contribution without testing it entirely. In 2016, 37% of the reviews were utACK whereas 32% of the reviews were ACK. Similarly, in 2017, only 24% of the reviews were ACK and more than 46% were utACK as the gap between ACK and utACK keeps growing over the years. The number of NACK reviews, or reviews that indicate that a pull-review should not be merged, is quite low every year, representing an average 6% out of the total or posted reviews. This may indicate that most of the comments that are not reviews, discuss and thoroughly work on the technical issues that may appear before being reviewed with a keyword. What's more, the average percentage number of nit reviews, or reviews that indicate that a pull-request should not be merged due to being trivial, is around 10%. Knowing this, we can conclude that a majority of reviews are positive. It would be interesting however to analyze how many ACK reviews are needed to approve a merge request in order to get merged.

We retrieved for Figure 4.15 the average number of reviews made for each merged

pull-request based on the keywords of Table 4.1. The Figure 4.15 shows the average for the pull-requests of the year 2015, 2016 and 2017.

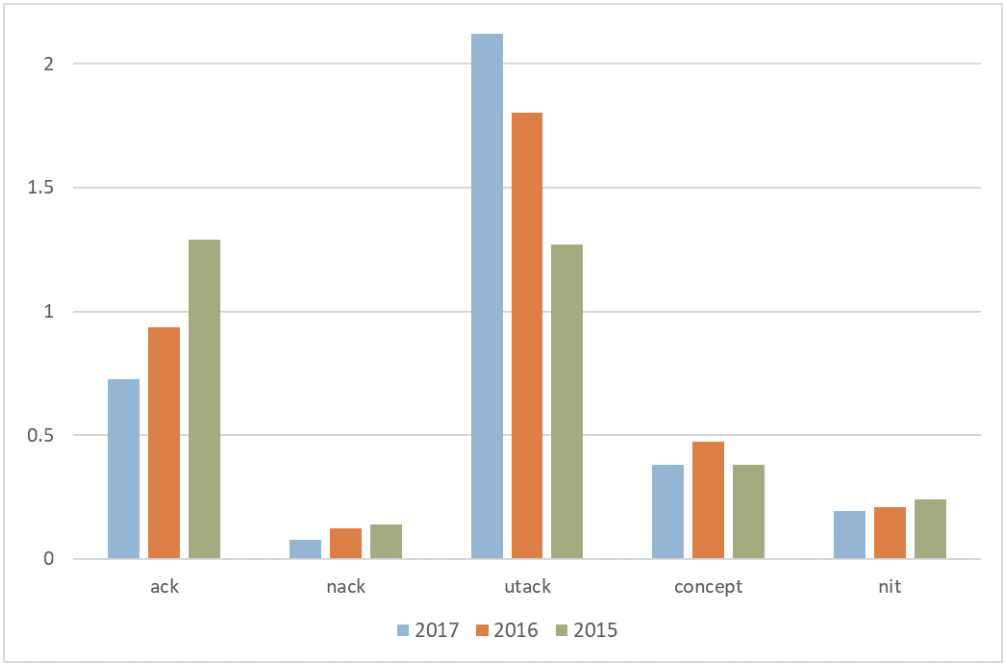


Figure 4.15: Analyzing the average number o*f Bitcoin Peer-review* keywords per merged Pull-request

In 2015, each merged pull-request had at least more than one ACK review, meaning that at least one developer took the time to test the pull-request and approve it. This number, however, kept decreasing over the year until reaching a ratio of 0.75 ACK review per pull-request. In contrast, as we also noticed on Table 4.2, the number of utACK reviews increases each year; in 2015, each merged pull-request had an average of one utACK review. But each year, this number increased until reaching an average of 2 utACK reviews per merged pull-request.

In conclusion, we can confirm the growth of the Bitcoin community and its participation in its development. Over the years, the number of users commenting and reviewing pull-requests kept growing. Despite a slow decrease of the number of contributors in the Bitcoin project, more community members are joining it and helping its development by reviewing code and changes made inside opened pull-requests. We notice however that these new community members and the comments they posts are usually for approving new changes, but they do not test the code thoroughly, only partially (confirmed by our utACK results from Table 4.2).

4.4 The economical aspect of Bitcoin

RQ4: *Did the introduction of the price factor have an impact on the continuity of its development?*

Since August of 2010, the Bitcoin token is available on non-regulated trading markets. This is an important milestone in the Bitcoin project as it gave for the first time a real-world value to this virtual currency. For over a year, since its creation during the mining of the genesis block, Bitcoin was simply a virtual asset traded among a very small community, mainly developers curious to this new technology. The project was previously seen as an open-source work, bringing innovative features such as the blockchain technology. However, the price introduction completely changed the view of this project, as it became a profitable virtual currency.

Figure 4.16 displays the Bitcoin price evolution every month since its first introduction on an exchange platform. The price is calculated based on the average of the Bitcoin price across major exchanges and trading platforms.

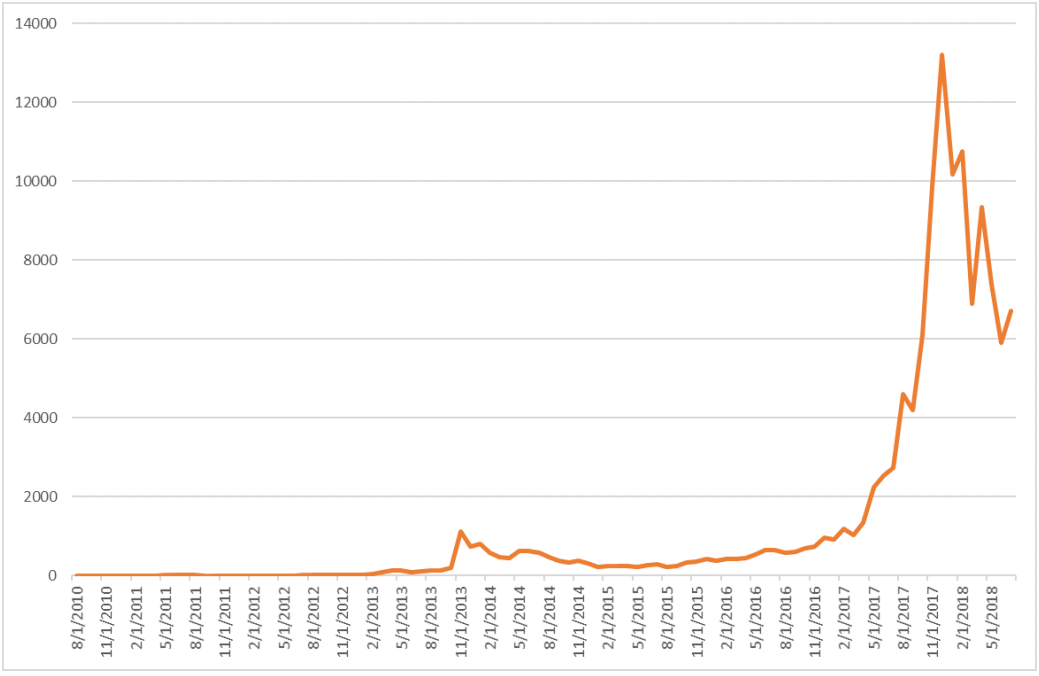


Figure 4.16: Bitcoin price every month

First available on the now-defunct exchange BitcoinMarket.com, the price went from 0.06$ in August 2010 to over 6,720$ in July 2018. This makes an increase of over 11,199,900% in just 8 years. We can, however, observe two specific times in the Figure 4.16 where the Bitcoin price reached a high point before decreasing at a lower level. In November 2013, the price suddenly peaked and reached an All-Time High (ATH) at over

1,120$. Back at that time, this increase was highly noticeable and impressive as the price of Bitcoin was previously quite low with an average of 32$ per Bitcoin before reaching that point. Another new All-Time High (ATH) was then reached on December 2017 at over 20,000$, but averaged down to 13,200$ in the Figure 4.16 due to its quick drop.

This ATH phenomenon happened twice by now and always has the same consequence; a huge drop in price afterward. In 2013, after reaching over 1120$, the price bottomed down to 226$ in 2015, making a decrease in price of around 80%. And again, in 2017, after reaching a top of 20,000$, the price immediately started to drop. We still cannot determine the bottom afterward as the correction is still ongoing. Nevertheless, if the pattern of 2013 repeats itself, we can expect a decrease of over 80% from the ATH which would put the price of Bitcoin at around 4,000$.

Taking into consideration the two bull-run events noticed on the Figure 4.16, the first one being between 2013 and 2015 and the second one from 2016 up until 2018 (as of August 2018, the correction of the second bull-run is still ongoing), we can use these two events and use them as a parameter to analyze the development behavior of Bitcoin.

Figure 4.17 merges both the Figure 4.13 and 4.16 to determine an eventual pattern. We limit however the scale of the graph between 2010 and 2016, right before the start of the second bull-run to better enhance the trend of the first Bitcoin All-Time High in price.

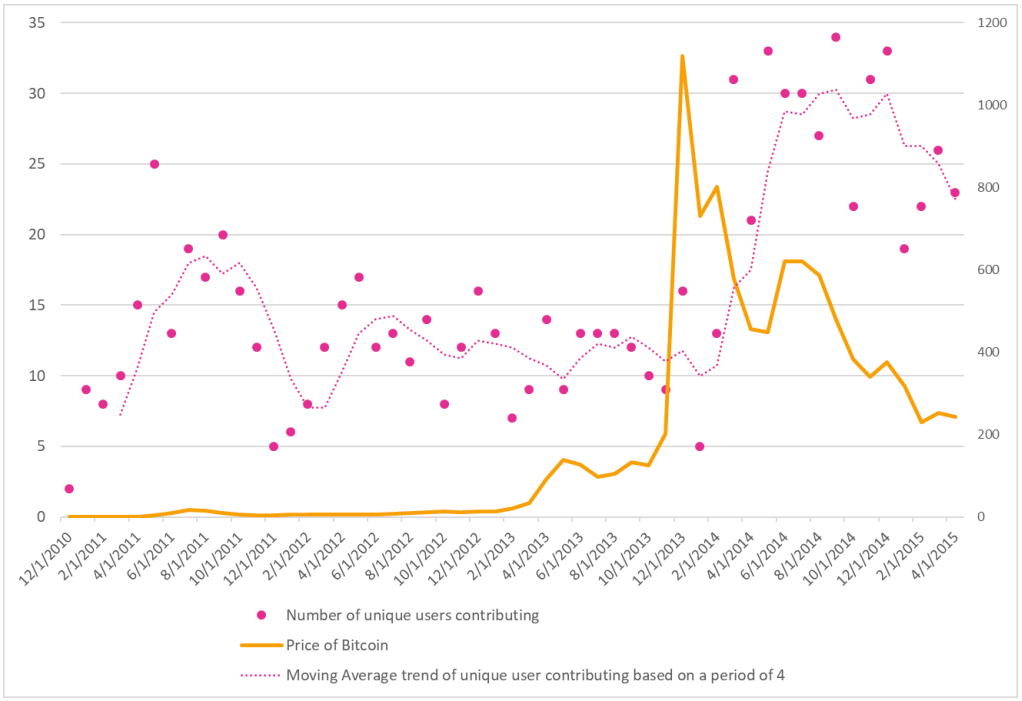


Figure 4.17: Number of unique users making commits to the Bitcoin project for each month paired with the Bitcoin price

We can immediately notice that after the price peak of Bitcoin and its following descent, the number of contributors increases. In fact, it increases from an average of 12 unique users contributing to the project each month between 2010 and 2014 to a high of 34 unique contributors per month. But after this increase in interest and users, as the price continues to fall down, the monthly number of unique contributors start to decrease. On March 2015, the price of Bitcoin felt to around 240$ and the number of unique contributors starts to decrease to 23, decreasing by 32% from its highest value in September 2014.

After visualizing the impact of the Bitcoin price into the Bitcoin contributor community, we will now do the same with the number of commits made monthly and see if the price of Bitcoin had any impact.

Figure 4.18 plots the number of commits made each month in the Bitcoin repository and compares it with the price of Bitcoin during the first Bitcoin bull-run between 2010 and early 2016. This separation between the first and second bull-run is necessary as the Bitcoin price increased exponentially afterward.

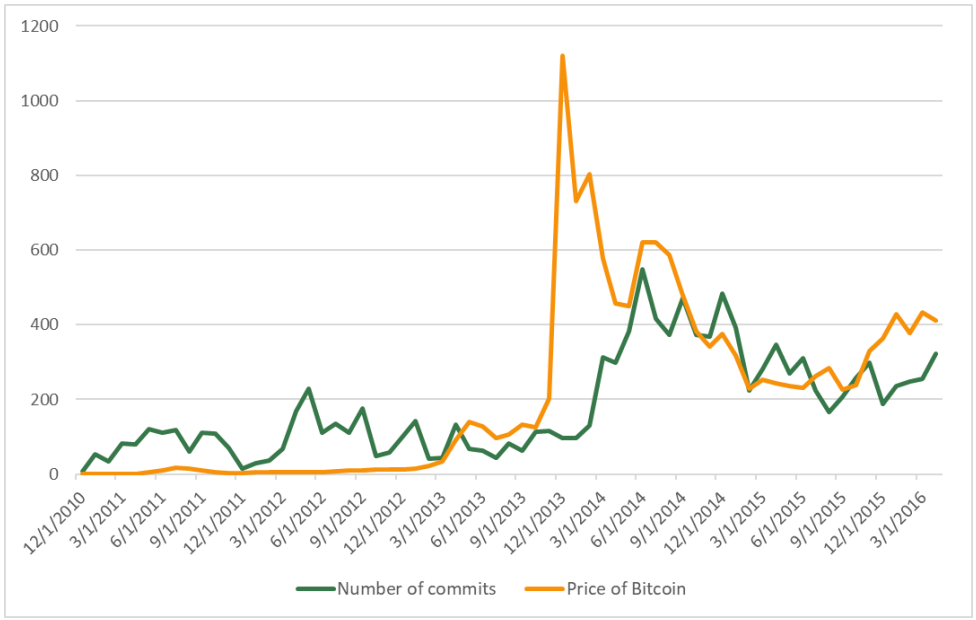


Figure 4.18: Monthly number of commits paired with the Bitcoin price during the first Bitcoin bull-run

The trend of Figure 4.18 is very similar to the to trend of Figure 4.17. After the huge price increase and Bitcoin, and its followed sudden decrease, the number of commits increased right after. The average before this movement was at around 88 commits per months, but quickly rose to a high of 549 commits during the month of June 2014. However, the number of contributions started to decrease, following the price decrease of Bitcoin. Nonetheless, the number of commits stayed at a higher level than before

with an average of 255 commits per months after its peak.

Figure 4.19 do a similar plotting as Figure 4.18, however, this graph will focus on the second Bitcoin bull-run that happened between 2016 and 2018.

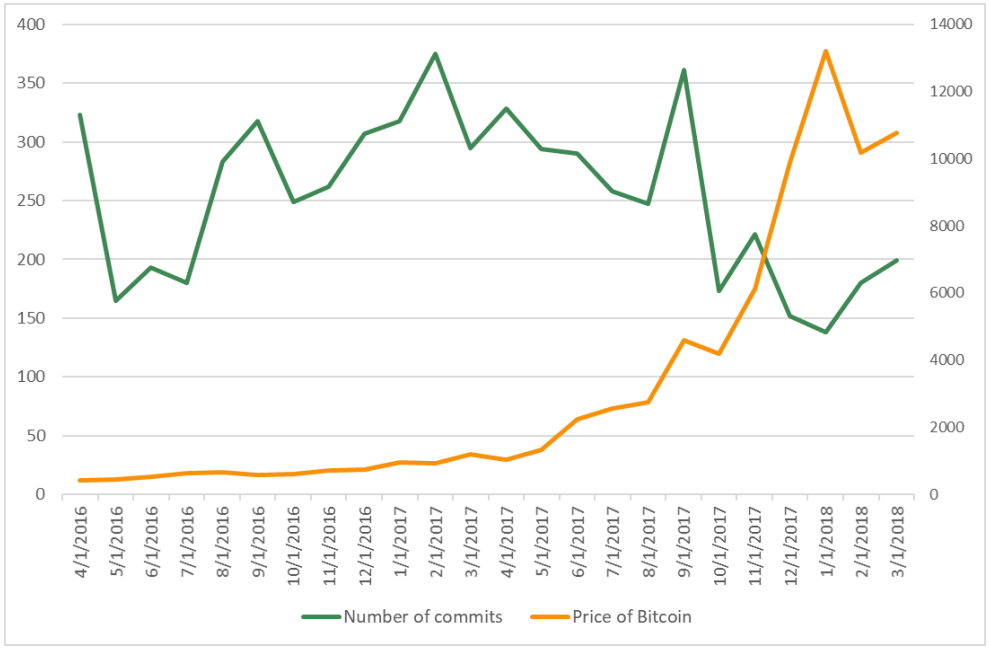


Figure 4.19: Monthly number of commits paired with the Bitcoin price during the second Bitcoin bull-run

During the second Bitcoin price bull-run, an inverse phenomenon happened; As the price of Bitcoin increased, the number of commits which was moderately high, suddenly dropped, going from an average of 280 commits per month to a lowest of 152 commits in December 2017.

Rather than having the number of commits made on the Bitcoin repository increase during its first bull-run, the second bull-run made the number of commits drop prematurely. We have to note however that as the time of writing this paper, the second bull-run has not finished, so the trend may change however the result is still the same as the price made the number of commits drop.

Through these observations, we can conclude that the price of Bitcoin has an impact on its development; because of the volatility of Bitcoin value, it created two bull-run events in which the price of Bitcoin suddenly increased at an exponential rate before suddenly dipping. The first bull-run generated an increase of contributors and contributions to the Bitcoin project, on the other hand, the second bull-run had the opposite effect, decreasing the number of contributions made to the project.

RQ5: *What is the relation between the Bitcoin price and the Bitcoin forks?*

We previously reviewed the impact of the price of Bitcoin on its development. However, an important aspect of open-source projects is the possibility to fork the project itself; duplicating its code in order to either work in a isolated environment, allowing anyone to freely apply any desired modification or change to the code, or work in a completely different direction, resulting in the split of the developer community. The Bitcoin project itself is a very interesting topic of analysis in this aspect as a fork of Bitcoin may be used for two purposes; either to work independently on the project, and merge afterwards the changes through a pull-requests (that needs to be approved by the community), or to create a whole new crypto-currency. In fact, by changing a few variables and restarting the mining on a new blockchain, Bitcoin can nowadays be used as a starting point for creating new digital currencies.

To identify if the price may motivate the creation of more Bitcoin forks, we create Figure 4.20 that shows the number of forks made from the Bitcoin repository and compare this data with the price of Bitcoin over time.

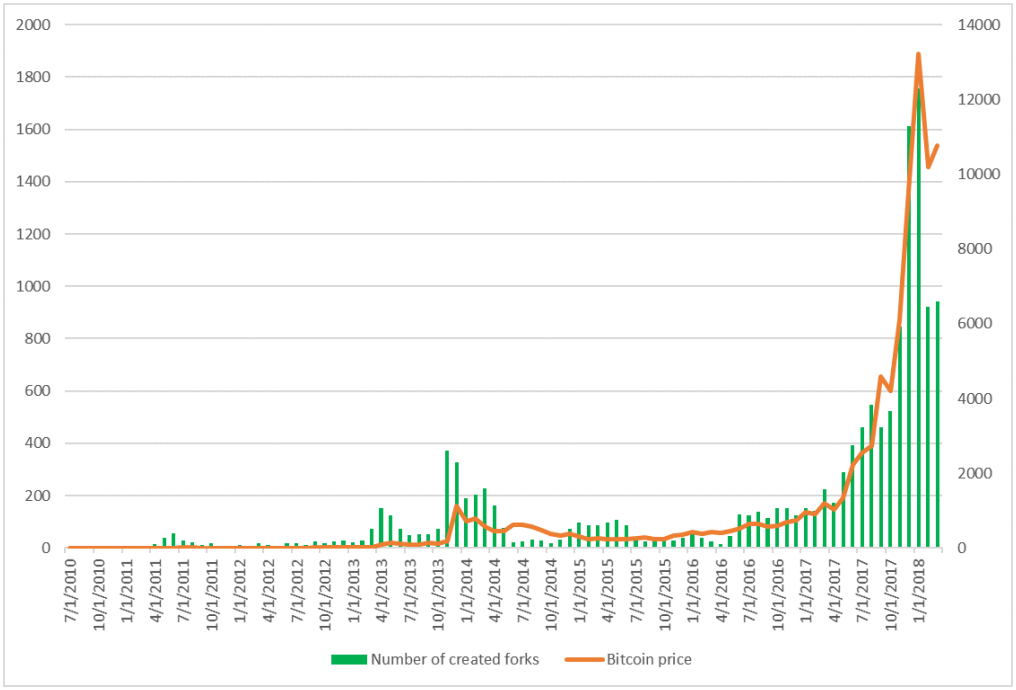


Figure 4.20: Number of Bitcoin forks created compared to the price of Bitcoin

What we can immediately notice is that the trend of Bitcoin forks created each month perfectly follows the trend of the Bitcoin price.

Out of these forks, we may want to see how many of them truly contribute and benefit the Bitcoin project. Figure 4.21 displays the previous number of Bitcoin forks created from Figure 4.20 and compares it with the number of forks that contribute and make commits to the Bitcoin repository project. We also computed the percentage of

contributing forks based on the total number of forks created each month and show its moving average of period 5.

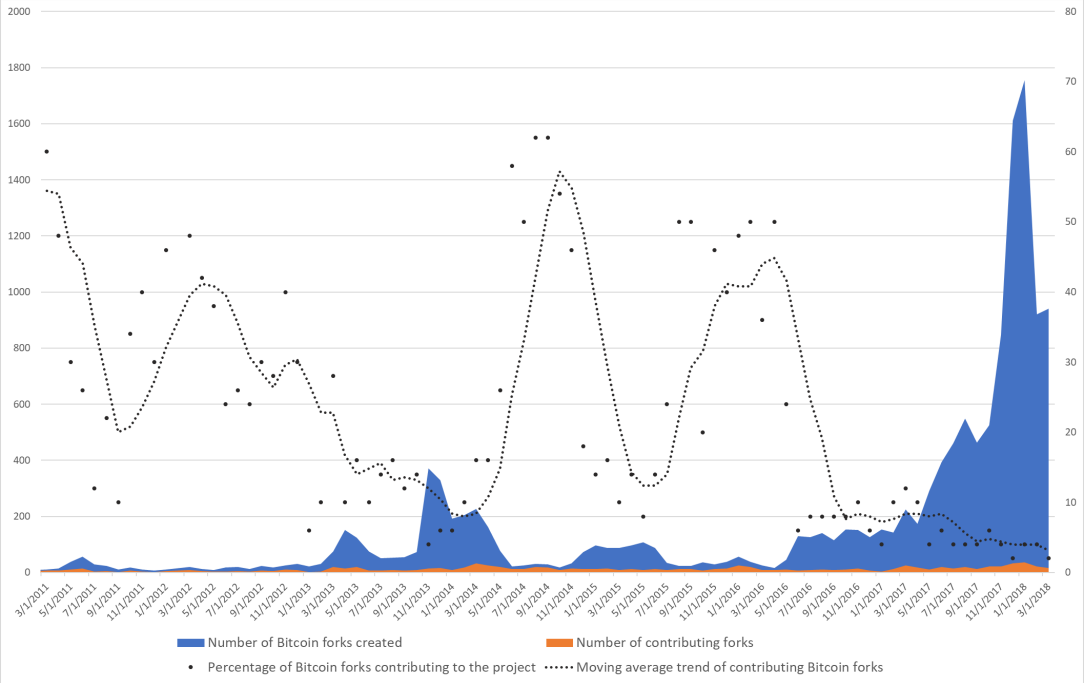


Figure 4.21: Number of created Bitcoin forks paired with the number of contributing forks and its contributing percentage

We notice that only a very limited number of Bitcoin forks are actually contributing to its original project. The highest number of contributing forks is 36 in January 2018, but this number pales in comparison to the total number of forks made during this month which is around 1,755. This indicates that only 4% of the Bitcoin forks created at that period were actually contributing. These contributing Bitcoin forks are all working through pull-requests, and are probably each owned by a single GitHub user as their private workspace environment. The number of contributing forks also correlates with the low number of unique contributing users that we analyzed in Figure 4.9. Furthermore, we can notice an inverse relationship between the total number of Bitcoin forks created and the number of these forks actually contributing to the Bitcoin project. We mainly notice two inversions of the trend, the first one between May of 2013 and June of 2014 and the second one between October 2016 until now. These two time-frames fit perfectly with the two bull-run events we previously discussed in Figure 4.16.

In conclusion, we discovered that the price of Bitcoin had a huge impact on the number of forks created; the number of forks perfectly follows the price of Bitcoin. This implies that the number of forks is greatly impacted by the price volatility of the Bitcoin currency.

4.5 Results overview

Table 4.4 summarizes the five research questions of this paper.

RQ	Question	Summary of the answers
1	*What is the process used to develop Bitcoin?*	The development process of the Bitcoin project has many similarities with a couple of open-source projects; it uses the BIP (Bitcoin Improvement Proposals) which are similar to Python's PEP (Python Enhancement Proposals), use an IRC channel and a mailing list in order to discuss ideas like Python and the Apache projects and attributes the role of "maintainer" to active contributors similarly to Python and the FreeBSD projects.
2	*How many people are working on the Bitcoin software?*	Since the beginning of the project, a total of 442 developers have contributed to the project. We noticed, however, a Power Distribution in which only a small number of people are doing most of the contributions. Similar to other open-source projects, 80% of the work is done by about 18 core-developers. This group of 18 people, however, evolves over time, as the core-developer team rarely stays the same each year. Furthermore, we noted a growth in the Bitcoin Organization every year, from 7 members in 2015 to 28 members in 2018.
3	*Is the Bitcoin open-source software an active community project?*	We determined that the Bitcoin open-source software was an active project. Over the years, the number of users commenting and reviewing pull-requests kept growing. Despite a small decrease of contributions in the project since its all-time high number of commits in 2014, the number of opened pull-requests kept growing. This indicates that even though not a lot of changes are made in the project, the community is still interested and participates in the implementation of new ones. And despite a slow decrease of the number of contributors in the Bitcoin project, more community members are joining it and helping in its development.

4	*Did the introduction of the price factor have an impact on the continuity of its development?*	The price of Bitcoin does have an impact on its development, especially during its two bull-run events in which the price of Bitcoin suddenly increased at an exponential rate before suddenly decreasing. The first bull-run generated an increase of contributors and contributions to the Bitcoin project, however, the second bull-run had the opposite effect, decreasing the number of contributions made to the project. At the time of writing this paper, the correction of the second bull-run has not finish yet.
5	*What is the relation between the price and the project forks?*	The price of the Bitcoin currency is strongly associated with the number of forks created; the number of forks perfectly follows the volatile trend of the Bitcoin price.

Table 4.4: Summary of the Research Questions

Chapter 5

Discussion

5.1 Summary of Key Findings

The purpose of the study has been to understand Bitcoin as an open-source project and to confirm if its price as a digital currency had an impact on its development. We have reported results relevant to each of our research questions to provide a better insight in this case study. A case study such as this one can provide important insights that could lead to testable hypotheses. Our results confirmed that there is indeed a relationship between the open-source development of Bitcoin and the price of the currency itself.

5.1.1 Bitcoin Development Process

To our knowledge, no other studies have examined the development process of Bitcoin like it is done in this study. Its development is fairly similar to other open-source projects, such as Python or on some aspects Apache. However, a general issue with case study papers is that they may not properly apply to other projects. Despite this, we can see that the Bitcoin project borrows many organization elements from other projects; most notably, the attribution of "maintainer" titles for active contributors to manage a specific set of functionalities in the project.

What's more, Bitcoin is an infamous open-source for its anonymous creator, Satoshi Nakamoto, who disappeared in 2011, without any warning. The project was left in the hands of its community, and they had the responsibility to ensure its continuity. This suggests that the community reviewed other open-source projects, in order to come up with a similar organization that would allow the project to continue despite the lack of a leader. Nevertheless, the Bitcoin community agreed to have a "lead-maintainer" that would serve as a leader; despite the decentralized community consensus in which the community has to approve any change or addition to the project, the maintainer will always have the last word. This may cause some issues as the community may not agree on the decisions taken by and the lead-maintainer and its team of maintainers.

Additionally, the title of "maintainer" is attributed to developers who often commit

to the project in a short span of time. The core-team would only take into account the contributor activity for a short amount of time and newly attributed maintainers could leave the project without warning. This issue happened with Andresen (also known on GitHub as gavinandresen) who was the first lead-maintainer of Bitcoin and got this title grandfathered from Satoshi Nakamoto as the latter disappeared. Andresen was one of the most active developers on the Bitcoin project, so his title was justified, however as he completely stopped contributing after 2014, he put the project in a complicated spot. Nonetheless, the community was reactive enough to elect a new lead-maintainer that would handle the continuity of the project, being Laan (laanwj) mainly because he was the most active developer at that time. Working in an open-source project means that the developers are contributing to the project without expecting to be paid, and rewarding the contributors with a title based on their merit is usually a method used to retain contributors.

What's more, the community of the Bitcoin open-source project came up with a very simple vocabulary to review changes or additions; because the project is a pull-request based development process, every addition or change has to be approved by the community. To simplify this review process, the Bitcoin community came up with a very simple Peer-review vocabulary that may originate from networking/interface protocol names. With the use of just a word, a developer can briefly explain his opinion in regard to a pull-request and whether or not it should be added or not. Because of its simplicity and popularity among Bitcoin contributors, the use of this vocabulary started to spread among other open-source projects as an easy mean to review.

5.1.2 The number of Bitcoin's developers

In this section, we measured the number of developers contributing to the Bitcoin open-source software in order to confirm if the development was as Satoshi Nakamoto envisioned; entirely decentralized. Our results demonstrated that the development of Bitcoin was spread in specific parts of the world and within a small number of developers. This indicates that the project is following the philosophy of open-source software development in which communities are made of loosely organized participants from all around the world.

Despite the development of the project not being centralized location-wise, Mockus, Herbsleb, and Fielding [2000] suggested that a core of 10 to 15 developers in an open-source project will control the code base, and will participate in 80% or more of the total contributions. Our results confirmed this statement on some aspects; while this may be true if we filtered the total number of developers, by taking into account every developer that contributed at least once, we can support the hypothesis that 18 developers are doing 80% most of the work. However, if we analyze the 18 most active developers each year, we can notice a strong change of core-developers each year. Only a small number of developers would contribute regularly each year to the project, as the other contributors would either leave or contribute less, creating a yearly shift in the core-developer team. But our findings on the Bitcoin organization on GitHub at least hint

that the number of core members in the project, despite starting low, grows over the years and entrusts responsibilities to new developers allowing the project to thrive in the long-term.

5.1.3 The Bitcoin community

The analysis of the community activity on the Bitcoin project is carried based on three factors; the contribution activity, the community interactivity and the pull-requests merging activity. These key-factors are chosen based on a methodology review from Eghbal [2018] who considers the approach of some researchers to be missing some important factors.

We first describe the results of the commits activity, which show that only a small number of developers are actually contributing to the project. Ever since the increase of both commits and contributors in 2014, the project has since struggled to stabilize. In fact, our results indicate a downtrend which would either implies that the project does not attract new contributors, or that the project became too complicated for developers to join in the middle of new feature developments.

Our second analysis was focused on the community activity, mainly inside the threads created for pull-requests. As we know, the Bitcoin project follows a pull-based development process in which each contributor works in a separate fork and asks for a merge once he is done. In order to merge the final changes, the community has to approve the pull-request created the by original contributor. This aspect of the project is important to review as maybe not many community members are participating, which would suggest that only a small number of people are deciding the integration of new changes to the Bitcoin-core. From our results, it is clear that the Bitcoin community is growing at a healthy pace. What's more, we confirmed that the number of comments posted was proportionate to the number of unique users participating. This indicates that despite a decrease of member committing the project, the community is still very active and participating, by reviewing and commenting on changes.

Our third approach consisted of analyzing the pull-requests activity. The results from the opened/merged/rejected pull-request quota activity analysis did not reveal any particular pattern, however, the results from our pull requests activity sampling analysis revealed that in the year of 2017, among the rejected pull-requests, an unusually large amount of additions and deletions were made inside of them. The fact that these pull-requests got rejected implies that they were bringing either unwanted changes or features that would stray away from the Bitcoin-core's community beliefs. Bitcoin is also a currency and any important update could have a moderate impact on the price of the virtual currency. In addition, implementation of new features from the core-development team requires a lot of planning beforehand, and the community cannot allow any unexpected upgrade of the codebase from the community.

Furthermore, extensive results carried out on the number of comment reviews made from 2015 to 2017 for each Bitcoin Peer-review keyword revealed a huge increase of "*utACK*" comment reviews, indicating that contributors are reviewing code changes

without actually testing them. This implies that these contributors reviewing these changes do not thoroughly test them; this could be either due to a lack of understanding the fundamentals of the Bitcoin codebase, or because the changes do not stray too much from the previous version of the code.

5.1.4 The impact of Bitcoin's price

Our findings indicate a relationship between the Bitcoin price and its development. The results of the Bitcoin price paired with different variables of the Bitcoin open-source development show that Bitcoin's price affects these variables. When the price of Bitcoin started to increase during the first bull-run, both the number of contribution and the number of contributors to the project increase as well. As expected, negative shocks of the price pushed the number of these contributions and contributors down. We believe the rise of the token's price increased the awareness of the project at that time. This may have presumably temporarily increased the number of developers which led to the increase of contributions.

However, our results demonstrated the opposite effect during the second bull-run, in which the increase in Bitcoin's price led to the sudden decline of Bitcoin's contributions. This suggests that the contributors had a reason to reduce their contributions or leave entirely the development of this project. Our study only focused on showing this phenomenon, hence we cannot answer why exactly this happened without speculating.

5.1.5 The relationship between Bitcoin's price and forks of Bitcoin

In addition to our findings from question 4, our results also demonstrated that the Bitcoin's number of forks was correlated with the Bitcoin's price. This result provides evidence of the hidden motivation from developers who wants to capitalize from the Bitcoin reputation by using the Bitcoin codebase to launch their own digital cryptocurrency.

Despite this, the other reason for these forks can come from developers' disagreements as the community can decide of the outcome of open-source projects. Bitcoin is no longer the only digital currency project currently worked on as new projects are made from forks of Bitcoin's codebase and go to different directions. And separate from what any one individual does or says, the majority always rule.

Our analysis of contributing Bitcoin's forks revealed an inverse relationship between the number of forks created from the Bitcoin project and the forks that are actually contributing to the original project. Taking into account the known time-frames of the two Bitcoin bull-runs, we can confirm the interest of developers to fork the Bitcoin project for their own purpose.

These results cast a new light on these forks; while some have been opportunistic attempts to make speculative profits, others are explicitly designed to address problems the designers see in Bitcoin, and thus often embed critiques of the Bitcoin model itself.

However, most of these alternative currencies do not represent a direct competitor to the Bitcoin. From the contrary, they can both cooperate in a mutually beneficial relationship; Bitcoin has benefited from being the first mover and increasing its network effect, but alternative currencies get the benefits of being able to start from a solid base and improve on it. When an alternative currency develops and integrates a new functionality, the others can benefit from it and fork it. This creates an environment of both competition and friendliness as every crypto-currency project can benefit from every single new update. Furthermore, as Bitcoin became more and more difficult to implement new features and changes to the code, autonomist communities have often retreated to small-scale projects.

5.2 Implications

In this section, we compare the results of our study to determine a logical explanation for the highlighted behaviors as it is worth discussing these interesting facts revealed previously. The results of our findings show clear evidence of the impact of the two bull-runs of the Bitcoin's price on its open-source development. During the first bull-run event, we demonstrated that both the number of contributions and contributors increased. This suggests that the rise of Bitcoin's price increased the awareness of the project to developers who were not aware of its existence. This statement is also confirmed by the positive relationship between the Bitcoin price and online searches from the findings of Kristoufek [2013] and Ciaian, Rajcaniova, and Kancs [2016]. Furthermore, thanks to the network effect established by the users and developers of Bitcoin, the project was able to establish a strong ecosystem which increased in popularity during this first bull-run.

However, as stated before, the second bull-run of Bitcoin's price had the opposite effect, causing a decrease in contributions during that event. At first, we could speculate that the interest on Bitcoin decreased in favor of the alternative crypto-currencies. However, when comparing our results from the high number of Bitcoin forks created during this time-frame and the considerable amount of additions and rejections inside rejected pull-requests in 2017, we can theorize that developers decided to focus development on forks instead of the Bitcoin. This could be due to:

- The greater flexibility available when working on another direction with a fork, where previously rejected changes on Bitcoin can be freely applied to the new fork's codebase.
- The financial value of the Bitcoin currency that allowed alternative crypto-currency to also serve as a speculative asset.

Based on these two factors, we could hypothesize that*:*

"A number of crypto-currency projects are created as a speculative asset rather than being a project for technological improvement".

Further research may consider using the findings of this paper and compare them with other open-source crypto-currency projects to confirm if the patterns between its development and its financial value are similar.

From Laanwj's perspective and current lead-maintainer of the Bitcoin project, "*Bitcoin infrastructure development must be one of the most hostile and crazy working environments in existence, at least in software development*"; this situation is mostly due to the fact that Bitcoin is nowadays more perceived as a speculative asset rather than an open-source software in the eyes of the public [Laan 2016]. As he describes it, the community can disagree on about every change made and "*affecting a whole economic system*" by their attacks. "*Some development tasks are extremely complex and require focus over a long time*" and what is more surprising from his statement is that "*the people attacking are, in many cases, not even users of the software*". This statement confirms that people are more fixated on the speculative aspect of Bitcoin rather than its development. This led to the fixation of the community on the implementation of changes that could eventually increase the price of the virtual currency instead of focusing on improving properly the software. Future research should consider the potential of investigating the BIP (Bitcoin Improvement Proposals) and their implementation in details that might reveal important findings in the Bitcoin contributor community and how their deployment might get impacted by the Bitcoin's price. They could also identify and analyze the community and developers sentiment and behavior through social networks to confirm their findings.

Because every update on the Bitcoin codebase requires a long time of planning, and due to the economic speculation that can impact the reception of a new change, developers may have preferred to stray away from Bitcoin when the price increased. This theory is in line with the sudden decline of contributions during the second bull-run of Bitcoin that we noticed in our results.

Furthermore, because the community is more focused on the price rather than the technology itself, it is not surprising to conjecture the rise of the number of forks with the increase of Bitcoin's price during the second bull-run. What's more, because of the previously mentioned long planning of every update by the Bitcoin developer community, it becomes more difficult for someone to join the development of Bitcoin. The project not only becomes more and more complex but some improvements may have already started their implementation after a long time of planning (IE. the Bitcoin Improvement Proposals) which may explain why some people do not interfere.

From Kolawa [2010]'s perspective "*adding developers to the project— which is probably the most common attempt to move a project along faster—actually decreases productivity*". This may sound illogical at first, however, the addition of new developers on any project usually hinders and slow the current developers of the project. The new developers may have a limited understanding of the current codebase and cannot effectively revise or work on the current code. What's more, their lack of familiarity with the current code may increase the chance of introducing new bugs to the code that will require more time to fix afterward. These findings lead to the conclusion that developers tend to work on forks of Bitcoin instead of the original codebase as it is less restrictive, and offers similar speculative opportunities. We can hypothesize that:

"The Bitcoin project's development became elitist and only a small number of core-developers are implementing the future improvements".

Finally, Mockus, Herbsleb, and Fielding [2000] hypothesized that "*In successful open source developments, the developers will also be users of the software*", however, due to the nature of the Bitcoin currency and the lack of regulations, some countries decided to forbid to the use of the currency in their territory. The most famous instance is China, who decided to rally against Bitcoin, causing a huge loss of volume and a drop in price. Because this currency cannot be used legally in that country anymore, it may seem risky for developers to continue working on this project. Depending on the ethics of the developers and their geographical location, a loss of Bitcoin contributions during the second bull-run may be linked to the ban of the virtual currency, who got highly targeted during that period. Based on this factor, we could hypothesize that:

"The viability of open-source crypto-currency projects are dependent of the developers' country legislation on these speculative assets".

Further research may take into consideration the political stance of crypto-currencies while reviewing the Bitcoin or other crypto-currencies development.

5.3 Limitations

Because of the potential and scope of the project, we decided to restrict our study to the Bitcoin project only. The findings that used GhTorrent may have a slight marge of error due to malfunctions in the GhTorrent mirroring system (software or network) that can result in some missing parts of the data. Analyzes concerning the community behavior inside pull-request comments couldn't be established due to the oversize GithubArchive database.

In addition, the analysis of the pull requests activity sampling had to be voluntarily restricted to the year 2015, 2016 and 2017 as the history retrieval of the Bitcoin project only started in 2015. Data concerning the year 2014 and before cannot be retrieved.

Furthermore, at the time of writing this case study, the second bull-run of Bitcoin has not finished its correction. The presented values, as well as hypotheses, may slightly vary once the Bitcoin's price has recovered. Another limitation of this study is the fact that only a global aspect of the Bitcoin project is analyzed and reviewed in order to identify some specific behaviors.

Chapter 6

Conclusion

The aim of this study was to better understand the nature of open-source development in crypto-currency projects and to gather results that could be used in future similar case studies. The findings of this study allowed us to understand the development process of the Bitcoin project, the activity of its community and the drivers behind its speculation. Analysis of the development has been presented as well as the community interactions. The development of Bitcoin adopts many operational processes from earlier open-source projects. Furthermore, we were able to find out that thanks to the network effect, the Bitcoin project has established a core group of both users and developers that can contribute to the project.

In terms of the impact of its financial aspect, we have identified two different patterns at two important moments of the Bitcoin currency. During the two bull-runs that happened in the Bitcoin's price, we observed two reactions in the development side of the project; during the first bull-run, we speculated an increase of awareness and interest in the project which raised the number of contributors and contributions in the project. Whereas in the second bull-run, we speculated a split in the Bitcoin developer community. In addition, the number of Bitcoin forks has a significant relationship with Bitcoin's price. Despite this, the relationship between forks of Bitcoin and contributing ones feel disconnected; the number of forks that contributes in the Bitcoin project stays low and stable throughout the years, yet the number of forks is volatile and disjointed to the number of contributing ones. The crypto-currencies are in a process of rapid development, and these non-contributing Bitcoin forks may illustrate that the forking of Bitcoin's code allows the creation of new crypto-currencies, with developers that may benefit from the general advance of the Bitcoin software and the speculative aspect annexed with Bitcoin's price.

This study's main contributions to the research on Bitcoin is the comprehension of the Bitcoin development as an open-source software and the identification of Bitcoin's price as a driver for its development trend. These findings should be taken into account in future research on Bitcoin and may help to analyze and review similar open-source crypto-currency projects based on the aforementioned findings.

Bibliography

Bornholdt, S. and K. Sneppen [2014]. "Do Bitcoins make the world go round? On the dynamics of competing crypto-currencies". In: *arXiv*.

Ciaian, Pavel, Miroslava Rajcaniova, and d'Artis Kancs [2016]. "The Economics of BitCoin Price Formation. Applied Economics". In: *Signaling theory: A review and assessment, Journal of Management*.

Eghbal, Nadia [2018]. "Methodologies for measuring project health". In: *Nadia Eghbal OSS notes*.

Farell, Ryan [2015]. "An Analysis of the Cryptocurrency Industry". In: *Wharton Research Scholars*.

Franco, Pedro [2014]. *Understanding Bitcoin: Cryptography, Engineering and Economics*.

Gousios, Georgios [2013]. *The GHTorrent dataset and tool suite*.

Grigorik, I. [2012]. *The Github archive*.

Hertel, Guido, Sven Niedner, and Stefanie Herrmann [2003]. "Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel". In: *Research Policy*.

Kolawa, Adam [2010]. "Software Quality and Software Development Productivity". In: *Parasoft - Software Development Management*.

Kondor, Daniel et al. [2014]. "Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network". In: *University of Maribor, Slovenia*.

Kristoufek, Ladislav [2013]. "Bitcoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era". In: *Scientific Reports volume 3*.

Laan, Wladimir J. van der [2016]. "Hostility, scams, and moving forward". In: *Laanwj's blog*.

Map Of Coins. URL: `http://mapofcoins.com/bitcoin`.

Martin, Guy and Gil Yehuda [2016]. "Building Leadership in an Open Source Community". In: *Linux Foundation*.

Matta, Martina, Michele Marchesi, and Maria Ilaria Lunesu [2015]. "The Predictor Impact of Web Search Media on Bitcoin Trading Volumes". In: *Information Filtering and Retrieval*.

Mockus, Audris, James Herbsleb, and Roy T. Fielding [2000]. "A Case Study of Open Source Software Development: The Apache Server". In: *Proceedings - International Conference on Software Engineering*.

Moon, Jae Yun and Lee S. Sproull [2002]. "Essence of distributed work: the case of the Linux kernel". In: *MIT Press, Cambridge*.

Nakamoto, Satoshi [2008]. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *bitcoin.org*.

Nyman, Linus [2015]. "Understanding Code Forking in Open Source Software: An examination of code forking, its effect on open source software, and how it is viewed and practiced by developers". PhD thesis. Hanken School of Economics.

Pirjan, Alexandru et al. [2013]. "Research issues regarding the Bitcoin and alternative coins digital currencies". In: Journal of information systems.

Polasik, Michal et al. [2014]. "Price Fluctuations and the Use of Bitcoin: An Empirical Inquiry". In: *University of Leicester*.

Raymond, E.S. [1999]. "The Cathedral and the Bazaar. Musing on Linux and Open Source by an Accidental Revolutionary." In: *Chicago Vol. 19, N2*.

Robles, G. and J. Gonzalez-Barahona [2012]. "A Comprehensive Study of Software Forks: Dates, Reasons, and Outcomes." In: *IFIP Advances in Information and Communication Technology*.

Schneier, Bruce [September 15, 1999]. *Crypto-Gram*.

Shah, Sonali K. [2006]. "Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development". In: *MANAGEMENT SCIENCE*.

Stallman, Richard [1994]. "Why Software Should Not Have Owners". In: *GNU*. URL: `http://www.ru.j-npcs.org/usoft/WWW/www_gnu.org/philosophy/why-free.html`.

Viseur, R. [2012]. "Forks impacts and motivations in free and open source projects". PhD thesis.

Weber, Steve [2004]. *The Success of Open Source*.

Young, Bob et al. [1999]. *Open Sources: Voices from the Open Source Revolution*.

Zhang, Yiteng [2014]. "Economics of Competing Crypto Currencies: Monetary Policy, Miner Reward and Historical Evolution". MA thesis. University College London.