ISSN 2278-3091

Volume 9, No.3, May - June 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse78932020.pdf

https://doi.org/10.30534/ijatcse/2020/78932020

Comprehending relationship between Contribution and Career development in OSS Ecosystem

M M Mahbubul Syeed[1], Kaniz Fatema[2], Razib Hayat Khan[3]
[1]Professional Member, ACM, Finland, mahbubul.syeed@gmail.com
[2]Professional Member, ACM, Finland, kaniz.fatem@gmail.com
[3]AProfessional Member, ACM, Bangladesh, razib.khan@aiub.edu

ABSTRACT

For the past couple of decades, Open source software(OSS) projects have become the pioneer in delivering commercial grade software products. Linux distributions, LibreOffice, Firefox, Chrome, Android OS initiative, FreeMind, BSD Distributions are few of the many such products. Unlike the in house commercial projects, OSS ecosystem offers a community driven collaborative development structure for people to freely join, learn, share effort and knowledge to develop software.

Due to this unorthodox process and practices, OSS Projects become the central concern of software engineering research. One of the widely researched topics on the track was what motivates developers to participate, how they get recruited and integrated in the OSS projects. However, the other half of the story remained unexplored. That is, being part of the projects, what factors related to developers' contribution lead to shape a sustainable and reputed career?

This research explores empirically the same by leveraging data extracted from OSS project repositories. The transcript of assessment shows that plurality, quality and length of service have strong positive correlation with developers' current rank and reputation. In other words, developers' reparation and acceptance within the OSS ecosystem grow proportionally with greater quality of contribution in diverse OSS projects for a substantial period of time.

Key words :Open Source Software (OSS), OSS Project, OSS Ecosystem, Contribution, Commit, Motivation, Career Development.

1. INTRODUCTION

A Software Ecosystem can be defined as *"A set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them*" [1][3].

Within the realm of Open Source Software (OSS) ecosystem this definition can be instantiated metaphorically. According to [6][2], OSS ecosystem consists of software projects that are freely contributed by the actors (e.g., developers, users, commercial stakeholders) within a shared market of mutual communication and dependencies. Often, these inter-dependencies occur due to free nature of resource flow. For instance, developers freely join or leave projects, can contribute in multiple projects simultaneously, and exchange views, knowledge and resources among them [6][7].

Apart from this, the OSS ecosystem consists of several complex artefacts that require a distinct characterization. Because these artefacts have vast contrast with their in-house commercial counterparts in several fundamental ways [2]. Therefore ,conventional software engineering knowledge fails to explain them. For instance, there are differences in process and practices(e.g., software production, services, project management, and market share), and in conceptual and structural particularities(e.g., principles, community participation, contribution, recognition and appreciation, governance, and legalities)[4][5].

The conventional concept of OSS process and practices dictates that an OSS project is often initiated by individuals(called owner/originator) and makes it openly available for anyone to contribute for further development. The owner sets the fundamental governing regulations regarding contribution, integration, management, and release planning [8].The usual goal of these projects is to create systems with open collaboration and access to support the pragmatic need of automation, rather filling up the commercial void [9]. However, due to their comprehensive and out-of-the-box solutions and quality, many open source products have not only penetrated the commercial market but also claim a large share of it[13].

In relation to conceptual and structural particularities, people in the role of active developer, peripheral developer, bug solver, bug reporter, reader, users [10][13] join or leave OSS projects freely without any legal bindings. Their contribution are voluntary in nature and without any monetary compensation. Therefore, it is difficult to reach consensus, and dictate developers to perform essential tasks, e.g., SQA or refactoring, and navigate the project development towards particular goals [11].

Despite of this unorthodox nature of OSS ecosystem, people rather proactively contributes to these projects, which in turn lead to huge market success [12][37]. Therefore, understanding the motives of the developers in contributing to OSS projects become one of the core concerns of software engineering research [13][9]. Consequently, empirical studies identified that the opportunity to learn and share knowledge in a highly professional environment [10], the pride to be recognized as a highly reputed developer within a massive community of professionals [13], personal satisfaction of accomplishment[11][15], and better job opportunities due to OSS experience and expertise [16][39] are the core motives.

However, the other half of the story remained unexplored. That is, being part of the projects, what factors related to developers' contribution lead to shape a sustainable and reputed career? This study carried out the same by leveraging the historical contribution records of thousands of OSS developers by mining OpenHub repositories [25][26].

This paper is organized as follows, in Section 2 research focus in the form of hypothesis is presented in relation to related background, Section 3 describes the data source, collection, cleaning and interpretation process. In Section 4 results are presented with statistical significance. Finally, future research directions, validity issue and concluding remarks are discussed in Section 5, Section 6 and 7, respectively.

2. RESEARCH FOCUS

One of the core research concern regarding OSS ecosystem is what motivates developers to join and contribute without any monetary compensation for a substantial period or time? [14][15]. Accordingly, research on understanding the process of software development [16], recruitment and integration of new developers into OSS projects has been conducted [13]. Findings suggest the followings.

First, OSS ecosystem offers a pragmatic environment for potential professionals and individuals to learn and share technology intensive knowledge [14]. They can learn about methods and principles of software design, development, tools, techniques and project management, by participating in specific projects or from project gurus [15][18]. Therefore, people took it as an opportunity to continuously improve or adapt their engineering knowledge and expertise freely being part of OSS ecosystem [16][20][21]. Moreover, OSS developers are often benefited with higher compensation packages and better employment in the industries, compared to their competitors who lacked in experience with OSS projects [19][22].

Second, with increased participation and consistent contribution in OSS projects, developers' get recognized as authentic, reputed and expert contributor within the ecosystem [19][23]. In OSS ecosystem, developers often send / receive token of appreciation (or credits) due to their valuable contributions. Reception of such credits from the peers often goes through a critical scrutiny of the contributions [22][38]. Therefore, accumulation of such appreciations, lead a developer to achieve their career pinnacle [32][21].

Third, developers in OSS ecosystem has the freedom to to participate in multiple projects simultaneously or at discrete point of time [19][24]. They took it an opportunity to extend their participation and contribution in multiple projects for greater exposure, acceptance and visibility. According to a survey, 5% of the OSS developers participate in 10 or more OSS projects [14].

Even though these studies empirically defined the core motivational factors of the developers in joining OSS projects, they failed shot to shed light on the next part of the story. That is, *being part of the ecosystem, what factors related to participation and contribution, assist a developer achieving his career pinnacle?* Because, statistics show that only around 10% of the active developers are able to achieve the rank of a top-notch developer during their career (see section 4.3).

Therefore, the core focus of this research is to empirically investigate the following,

What are the key factors related to participation and contribution in OSS projects that assist developers to attain a sustainable and reputed career within the OSS Ecosystem?

Under the hood, four distinct hypotheses are formulated and examined by mining data from OSS project repositories. These hypothesises will evaluate four distinct factors associated with participation and contribution and are analogous to the motivational perspective of developers (as discussed above). Statistical verification of the outcomes for theses hypothesis, and a cumulative assessment of the result will draw a holistic view on the topic. In what follows, the definition of the hypothesises.

H1:*Plurality of contribution is positively associated with the developers' reputation in OSS ecosystem.*

Contributing in multiple projects simultaneously or at discrete point of time is popular among the OSS developers and is a common practice. This hypothesis therefore investigates the relationship between plurality of contribution in terms of multiple projects to that of the developers' current reputation and rank.

H2:*Persistent participation in projects is positively correlated with higher reputation in OSS ecosystem.*

Many developers have long career trail in participating different OSS projects. For long lived developers this trail ranges from ten to twenty-five years of contribution. Therefore, it would be interesting to verify the impact this persistent participation over the current reputation of the developers.

H3:*Trait of contribution is positively associated with higher reputation in OSS ecosystem.*

It is a well-known fact that developers contribute in OSS projects at different capacity, which brings a huge contrast on the quality of what they contribute. Traditionally, developers in OSS projects often review and assess the contribution of their peers, and in turn recognize the highly significant contributions. This process distinguishes the quality contributions and their contributors. Therefore, this hypothesis is intended to verify the correlation between the quality of the contribution and the reputation of the contributing developers, which is assumed to be positive.

H4:*Volume of contribution should have a correlation with higher reputation in OSS ecosystem.*

As per the conventional wisdom, higher exposure in career should be due to greater contribution in service. Therefore, the more a developer contributes (in any valid category), the better it is for his career growth. This hypothesis thus intended to verify the relationship between the total number of contributions to that of a developers' current reputation.

3. RESEARCH APPROACH

3.1 Data Source

OpenHub [25] is one of the largest repositories of OSS Project data, and are widely used for research now-a-days. Open-Hub, since its' inception has consistently built a huge repository of data by connecting 5,062,968 open source contributors, is extracting 1,319,771 version-controlled repositories (e.g., CVS, SVN, Git, Bazaar, and Mercurial), indexing 496,022 OSS projects, and analysing 29,170,080,853 lines of code [25].

Alongside, statistics on OSS projects, people, artefacts and ecosystem are also derived and made available for study. For instance, statistics of developers' activities, commit records, length of contribution, language expertise, list of OSS projects contributed, social interaction record and others [25].

All this information is made freely available by OpenHub for research through a number of open APIs [26]. This study extracted all the related information from OpenHub to evaluate the hypothesises.

3.2 Terms and Interpretation

Following standard terms are used in this study which carry their usual meaning according to literature and OpenHub.

Contribution in OSS projects refer to any form of commits (e.g., code, comment, bug report, patch, documentation) made by the developers. Merit of a contribution is judged against several factors, e.g., severity of the problem solved, code quality, etc.

Kudo is a social network-based concept adopted by OpenHub to send and/or receive token of appreciation from peer experts within the OSS community [25]. Depending on the merit of the contribution, a developer may receive kudos from other developers. Such appreciation (aka Kudos) are accumulated to rank and distinguish developers (e.g., core and periphery developers) [27].

Kudo Rank defines the rank of an OSS developer on a scale of 1 to 10, which is calculated by leveraging kudo send and receive records [27]. Here, 1 is the lowest rank and 10 is the highest. A kudo rank is the product of continuous assessment of a developers' contribution over a period of time [28], and is associated with his reputation, skill, and productivity within the community [27][28].

3.3Data Collection

OpenHub data are free to download with appropriate API calls, which return results in XML format [26]. Prior to use the APIs, an API key has to be accrued from OpenHub admin through formal application [26]. Once the key is accrued, a Java program is written and executed to automate the API calls with the given key. Received XML data are stored in text files for further processing. A brief description of the XML data is presented bellow.

Member Account information. This study extracts 563,427 registered members (otherwise developers) account information from OpenHub repository [26]. Following information related to a member account is initially collected, Account id: the unique id of the registered member.

- Name: the name of the account older.
- Post count: total number of posts made by a member in OpenHub forum.
- Kudo score badges: this badge holds two distinct information, namely, KudoRank and position. This study utilizes the KudoRank as defined before.

Contributor Information. A contributor dataset holds project specific contributions made by each member. For this study, a total of 844,012 contribution records is harvested, each of which holds the following data items:

- Contributor's id and name (these two fields are specific to the project in which the contributor is contributing).
- Project id in which contribution is made.
- Comment fraction in a commit.
- The time when a contributor commits his first and the last contribution.

Total number of commits and the man month (i.e., the number of months for which a contributor made at least one commit). In the context of OpenHub Man Month represents the accumulated contributions made by a developer in OSS projects, e.g., LoC committed, documentation, writing project wiki, and mailing list activity.

Kudo Received History. Following information related to kudo received by a member for a given period of time is collected,

- Sender account id and name.
- Receiver account id and name
- Project id and name
- Contributor/receiver id, name and date when the kudo was received.

A total of 46,926 kudo received entries are collected for this study.

Kudo Sent History.Following information related to kudo sent by a member to his peer is collected,

- Sender account id and name.
- Receiver account and name.
- Project id and name.
- Contributor/receiver id and name, and date when kudo was sent by the sender.

For this study 57,458 kudo sent records are collected. Kudo sent and received history data are then combined to generate a uniform dataset that has the following data items: Sender account id and account name, receiver account id and name, project id and name to which the kudo was sent, contributor id and name, and the date when kudo was sent.

3.4Data Processing

Data processing activity involves getting rid of the irrelevant and missing information, eliminate outliers, and clustering the data in accordance with the hypotheses. The processing stage executes the following actions in sequence,

- A JAVA application is executed to extract the data from XML files and saved in XLSX format. In this phase, data associated with each XML tag is extracted to avoid any accidental data loss. The generated XLSX file holds raw data where each data column corresponds to an XML tag.
- From the raw XLSX file, the data columns that are pertinent for assessment of the hypothesises are retrieved and clustered into individual files, one for each hypothesis. For instance, data columns listing total number of projects contributed by the developers are extracted to separate file to validate Hypothesis H1. Once done, an automated checking and cleaning of null values is carried out using macros.
- Finally, the clustered data are aggregated, and statistical validation (e.g., min, max, avg, correlation, regression as appropriate) are carried out to evaluate the hypothesises.

Even though the data used for this study is protected by the non-disclosure agreement of the conducting institution, the research makes the partial snapshot of each of these steps available in the given link [29].

4. ANALYSIS AND SYNTHESIS

4.1Plurality of contribution is positively associated with the developers' reputation in OSS ecosystem (H1).

Plurality is defined by the number of OSS projects contributed by a developer, either simultaneously or at different time periods. Within the OSS ecosystem, joining and contributing to multiple OSS projects are a common phenomenon. According to a survey, more than 5% of all OSS developers contribute to10 or more projects [17]. Therefore, investigating the correlation between the plurality of contribution to that of the rank of a developer might reveal significant insight.

In order to do this assessment, the average number of projects contributed by all the developers at each rank is calculated, which is given by the following formula, assume that

$projects_j$is the no of projects contributed by d*eveloper*$_j$

n is the total number of developers at $Rank_i$

$avg_i$is the average no of projects at $Rank_i$.

$$\textbf{\textit{Then}},\ avg_i = \sum_{j=1}^{n} Projects_j / n$$

The value of avg_i is plotted against the corresponding $Rank_i$, where $i = 1 \text{ to } 10$, in a bar chart (Figure 1). According to this chart, developers at the top rank (i.e., Rank 10)contributed to 11 projects on an average. This is the highest among all the ranks. Analogues to this, developers at rank 9hold the second position in terms of number of projects contributed. These outcomes might indicate towards a linear association between the rank and the project count; however, it is not. For instance, developers at ranks 1, 2 and 3 perform almost as good as the developers at rank 4, 5 and 6, respectively, in terms of number of projects contributed. Additionally, developers at rank 6 has contributed to equal number of projects(in average) to that of rank 9 developers.

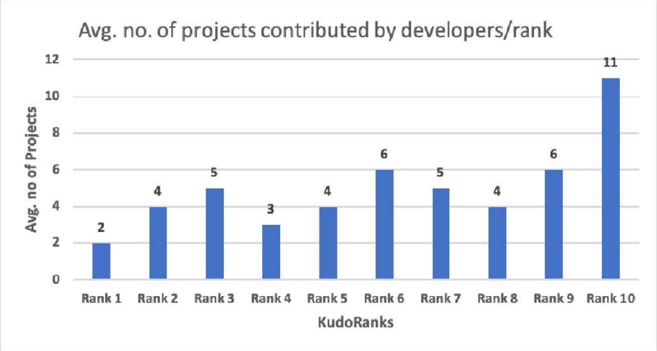


Figure 1: Average no of Projects Contributed by Developers in each Rank

Therefore, in order to better understand the significance of plurality of contribution (i.e., avg. no of projects) over the rank, this study further measures the Pearson correlation coefficient(r) with the probability of significance p-value. The result shows,

Correlation coefficient $r = 0.7341$ with
Probability of significance, $p\text{-}value = 0.0156$.

For reference, a *r value closer to +1 (plus one)* shows strong positive correlation between the data sets and a *p-value* <*.05* denotes that the correlation is statistically significant and reliable. Therefore, there is a strong and significant positive correlation exists between the rank of the developers and the plurality of contribution (in terms of total number of projects contributed) and thus the hypothesis holds.

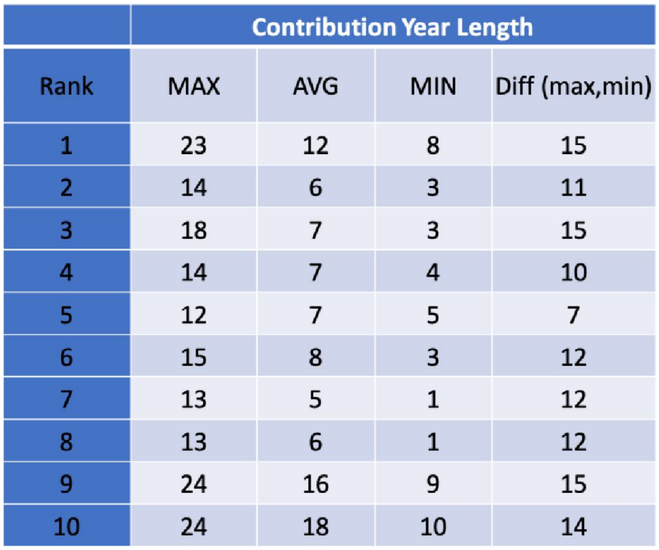
However, correlation does not define causality (i.e., cause and effect) relationship between the data sets. Therefore, it cannot be claimed conclusively that plurality is the only factor related to project contribution that defines developers' rank. There may be other associated project factors that also have impact in this regard. This reasoning is also evident from the chart data, where lower rank developers found contributed to same number of projects as that of the higher ranked developers. This research identifies other project related factors (e.g., project profile, rating, impact on the market) that might play apivotal role in defining the rank.

4.2Persistent participation in projects is positively correlated with higher reputation in OSS ecosystem (H2).

In this research *Persistent Contribution is defined as the total number of uninterrupted years of participation of a developer in OSS projects*. In OpenHub, this length of participation is calculated from the date of their first commit to to-date [26].

This measure is an essential factor, because developers who are motivated to enhance their career, have to live long in service [15][18]. Only then, they can consistently learn, develop expertise, and contribute effectively in the projects, which in turn will lead to their career pinnacle [21].

In order to analyse this perspective, the total year of contribution for each developer in each rank is tabulated. Then for each rank, the maximum, minimum and average number of years of contribution is calculated. Additionally, the deviation between the maximum and minimum for each rank is computed. These data are tabulated in Figure 2.



	Contribution Year Length			
Rank	MAX	AVG	MIN	Diff (max,min)
1	23	12	8	15
2	14	6	3	11
3	18	7	3	15
4	14	7	4	10
5	12	7	5	7
6	15	8	3	12
7	13	5	1	12
8	13	6	1	12
9	24	16	9	15
10	24	18	10	14

Figure 2: Contribution Year Length from first commit

The very first observation that can be drawn from this tabulation is that developers in the lowest rank (i.e., in Rank 1) has one of the highest years of participation for both maximum (23 years, which is the 2nd highest) and the minimum (8 years, which is the 3rd highest). Alongside, the average value (which is 12 years) suggests that the two extents are not outliers and most of the developers have around 12 years of participation record. This observation raises an immanent concern, *despite of such persistent participation in OSS projects, why are these developers in the lowest rank?*

Related research on the track suggests that within the OSS ecosystem developers can be classified in different categories depending on their participation, which is popularly defined as onion model [30][31]. Within this model, the active developers, bug solvers reside around the centre, and the bug reporter, peripheral developer, reader, and passive users reside towards the outer surface. Most of the developers who are merely readers and passive users have long participation records with shallow contribution in the projects. They mostly remain with the projects to gain knowledge and get the updated product to be used for their need [32][30][33]. Therefore, this rank is considered outlier for this investigation.

For the other ranks (ranks between 2 and 10), it can be noted that the top ranked (rank 10 and 9) developers have the highest number of participating years (in all three measures). This sets a baseline in terms of persistence of participation to attain higher ranks. However, in order to verify this opinion, Pearson correlation coefficient is calculated between the average length of contribution and the corresponding ranks. The result shows,

Correlation coefficient $r = 0.70$ with

Probability of significance, *p-value* = *0.04.*

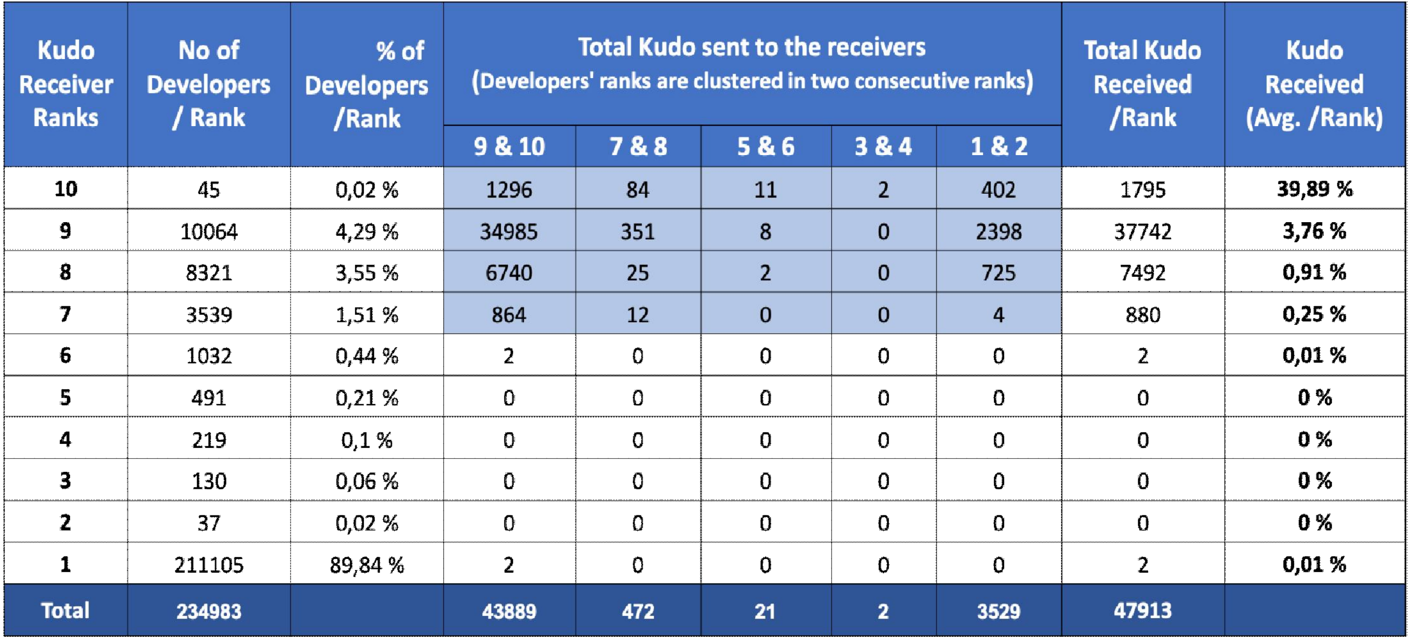
As both these values are statistically significant ($r \cong +1$ *and p-value* < *.05*), therefore, it can be asserted that the rank of a developer is positively correlated with their persistent participation in OSS projects.

However, the variance between the max and min years of participation for the higher ranks (e.g., ranks between 7 to 10) raises further concern to be investigated. Because, year difference is 14, 15, 12, and 12 for ranks 10, 9, 8 and 7 respectively (Column 5, Figure 2). This difference is exceptionally high for developers who are at the same rank. This research recognizes few tightly related factors that might explain the underlying reason of this variance. For instance, developers' consistency of contribution in terms of man-month (that is the total number of consecutive months in which a developer has made at least one commit), the quantity and quality of contributions.

4.3Trait of contribution is positively associated with higher reputation in OSS ecosystem (H3).

Developers in OSS projects contribute in different capacity, which ranges from serious code contribution, bug resolution, bug reporting to simple documentation, and merely being passive observers [32][21]. As such, their trait (i.e., quality) of contribution varies widely. Therefore, in order to appreciate great and invaluable contributions done by the developers, a method of appreciation is practiced with OSS ecosystem. This is often known as Kudo, which is a token of appreciation send to a contributing developer by others' in the community to acknowledge their scholarly effort. Accumulation of such kudos lead a developer to achieve their career pinnacle (i.e., higher kudo ranks) [22].

In this study, *trait of contribution* is measured from the perspective of the top-notch developers who sent kudos to others. The educated assumption here is that the top-notch developers definitely assess the quality of contribution (i.e., commits) before attributing kudos against it. Because, these developers have comprehensive knowledge on the projects, in-depth technical know-how, and analytical ability to assess merit of an issue and its' resolution. Therefore, they are the best people to attribute kudos to contributing developers in an authentic way.



Kudo Receiver Ranks	No of Developers / Rank	% of Developers /Rank	Total Kudo sent to the receivers (Developers' ranks are clustered in two consecutive ranks)					Total Kudo Received /Rank	Kudo Received (Avg. /Rank)
			9 & 10	7 & 8	5 & 6	3 & 4	1 & 2		
10	45	0,02 %	1296	84	11	2	402	1795	**39,89 %**
9	10064	4,29 %	34985	351	8	0	2398	37742	**3,76 %**
8	8321	3,55 %	6740	25	2	0	725	7492	**0,91 %**
7	3539	1,51 %	864	12	0	0	4	880	**0,25 %**
6	1032	0,44 %	2	0	0	0	0	2	**0,01 %**
5	491	0,21 %	0	0	0	0	0	0	**0 %**
4	219	0,1 %	0	0	0	0	0	0	**0 %**
3	130	0,06 %	0	0	0	0	0	0	**0 %**
2	37	0,02 %	0	0	0	0	0	0	**0 %**
1	211105	89,84 %	2	0	0	0	0	2	**0,01 %**
Total	**234983**		**43889**	**472**	**21**	**2**	**3529**	**47913**	

Figure 3: Kudos' received by developers of Each Rank during their career in OSS ecosystem. Kudos are sent by other developers to appreciate ones' contribution.

In order to investigate this association between the quality of Kudos received and the corresponding rank of the developers, this research categorically present the relevant data with relevant analysis in Figure 3. The reported data is generated by evaluating the kudo send and receive records of 234,983 developers.

The first column in Figure 3 lists the receiver developers' kudo ranks in a descending order. Corresponding rows in the second and third column present the total number of developers and their percentile in a given rank, respectively. This simple statistic reveals that only 45 (.02%) developers attained rank 10, 10064 (4.29%) developers attained rank 9 and 8321 (3.55%) developers attained rank 8, out of more than 234K registered developers. On the contrary, almost 90% of the developers are at the primitive stage (i.e., Rank 1) of their career. Thus, reaching the pinnacle in ranking is not a trivial task to accomplish for the developers.

Further investigation calculates the total number of kudos received by the developers in each rank and listed under the ranks of the sender developers. This data is registered in column 4 through column 8 in Figure 3. For example, developers at rank 10 received a total of 1296 kudos from rank 9 and 10 developers, 84 kudos from rank 7 and 8 developers, 11 kudos from rank 5 and 6 developers, and so on. These classifications of kudo according to the rank of the senders would facilitate identification of several quality perspectives, which are reported next.

According to this kudo sent record, the top ranked developers (rank 7 through 10) received 47,909 number of kudos (which is around 99.9% of the total kudos sent to all developers). These kudos are sent by developers of all ranks (see the highlighted cells of top 4 rows in Figure 3). Among the senders, only the top ranked developers with ranks 7 to 10, sent 44361 number of kudos (which is 92,58% of the total kudos sent) to the receivers. This cross-sectional perspective of kudo sends, and kudo received with respect to developers' rank leads to the fact that kudo send and receive activities are predominantly performed only by the top ranked developers. Therefore, the following observation can be ascertained,

The scholarly contributions made by the top-notch developers are rigorously verified by their counterparts of similar ranks before attributing the kudos.

Finally, this study calculates the per capita kudo received by each developer per rank. The result is shown in the last column of Figure 3. According to this statistic, a developer at rank 10 received 39.89% of the total kudos sent, followed by the rank 9 developers (3.76%). Developers of other ranks have very insignificant portion of kudos received (either bellow 1% or 0). This statistical evidence strengthens the prior assertion that the rank of the developers is positively associated with the trait of contribution.

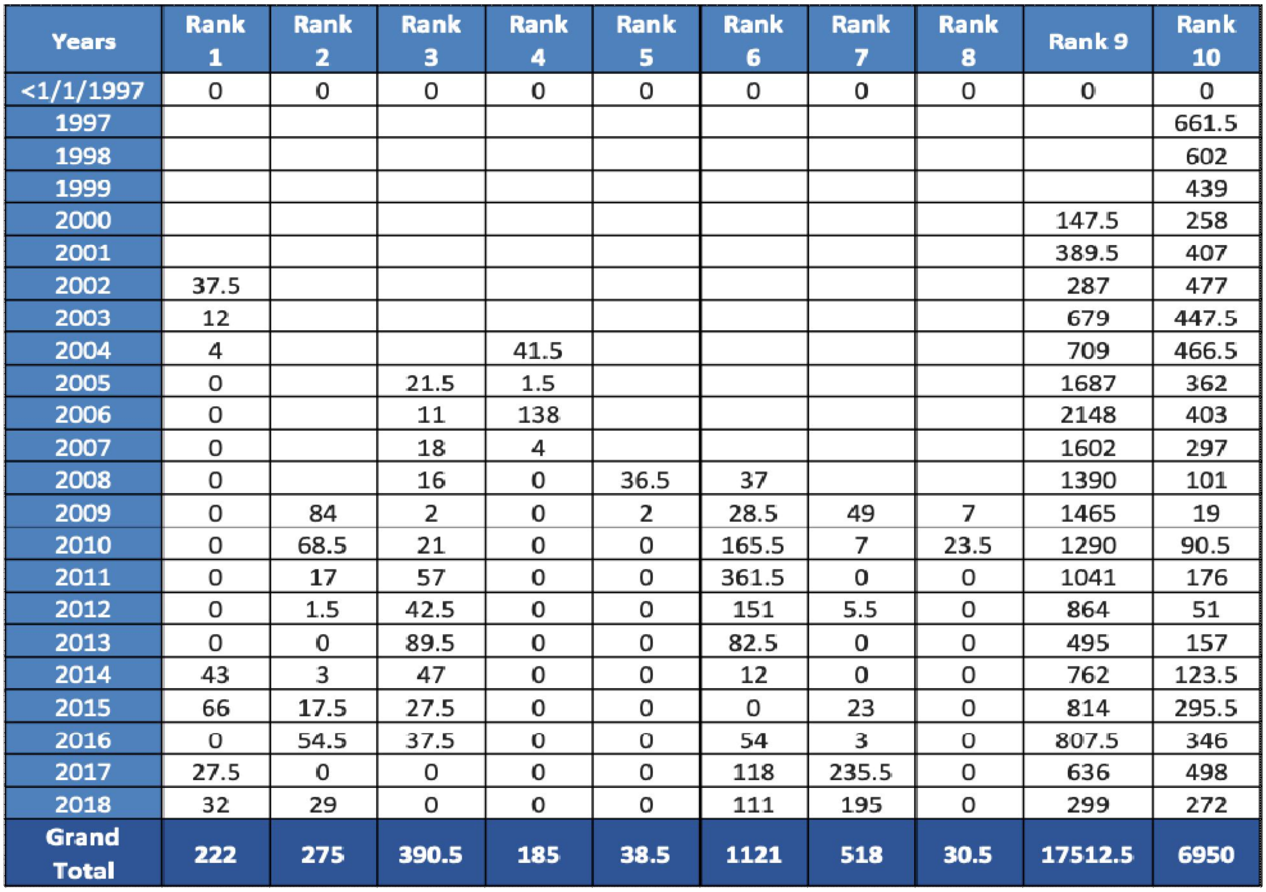
Furthermore, the *Spearman correlation coefficient* on these data (data in column 1 and 10, Figure 3) reports the following,

Correlation coefficient $rs = \rho(rho) = 0.82$ with
Probability of significance, $p\text{-}value = 0.004$.

The correlation shows strong positive association between the data with high certainty. All these evidences lead to proclaim that the hypothesis holds.

4.4 Volume of contribution is correlated with higher reputation in OSS ecosystem (H4).

Conventional wisdom suggests that the more a developer contributes to the projects, the more they will gain visibility, appreciation, and get promoted to higher ranks. This hypothesis is posed to verify the same by evaluating the relation between quantity of contribution and the rank of the developers [36].



Years	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10
<1/1/1997	0	0	0	0	0	0	0	0	0	0
1997										661.5
1998										602
1999										439
2000									147.5	258
2001									389.5	407
2002	37.5								287	477
2003	12								679	447.5
2004	4			41.5					709	466.5
2005	0		21.5	1.5					1687	362
2006	0		11	138					2148	403
2007	0		18	4					1602	297
2008	0		16	0	36.5	37			1390	101
2009	0	84	2	0	2	28.5	49	7	1465	19
2010	0	68.5	21	0	0	165.5	7	23.5	1290	90.5
2011	0	17	57	0	0	361.5	0	0	1041	176
2012	0	1.5	42.5	0	0	151	5.5	0	864	51
2013	0	0	89.5	0	0	82.5	0	0	495	157
2014	43	3	47	0	0	12	0	0	762	123.5
2015	66	17.5	27.5	0	0	0	23	0	814	295.5
2016	0	54.5	37.5	0	0	54	3	0	807.5	346
2017	27.5	0	0	0	0	118	235.5	0	636	498
2018	32	29	0	0	0	111	195	0	299	272
Grand Total	222	275	390.5	185	38.5	1121	518	30.5	17512.5	6950

Figure 4: Yearly total commits per Rank

However, in knowledge intensive jobs it is often the quality of contribution that matters more than the quantity. According to the famous art critic and philanthropist, John Ruskin

Quality is never an accident. It is always the result of intelligent effort. Software development is a knowledge intensive engineering task [34].

This is often done by applying verified engineering knowledge complemented by prior experience, and problem-solving capability of the developers to get the task done with quality assurance [34][35]. Accordingly, there are two possibilities exists in achieving the career pinnacle:

a.Quality of contribution is what matters than the quantity, or

b. Both quality and quantity of contribution matters.

This study has already established a strong correlation between the quality and the rank in evaluating hypothesis H3 (in Section 4.3).

Therefore, in order to explore the relationship between the quantity of contribution and the rank, developers' total commits per year is counted and grouped by each rank. The data is tabulated in Figure 4. Here, the columns correspond to the ranks and each row represents total number of commits made by developers of each rank for a given year. The year is starting from 1997 in which the first commit was recorded by the OpenHub repository. Note that this table records Zero commit entries for several years and for multiple ranks (e.g., for rank 1 there is no commit recorded for the years 2004 through 2013). This void in record may be due to missing data in the source repositories. As yet it possesses a validity threat towards the observation drawn next.

Based on the available data, it can be observed that developers belong to top ranks (e.g., Rank 10 and Rank 9) made consistently high number of commits from the year of their commencement. Albeit, developers in lower ranks (e.g., rank 6 and rank 3) also made considerable number of commits, which in some cases surpass the commits of rank 10 developers. For example, in the years between 2010 and 2012, developers in rank 6 made higher commits than developers in rank 10. This contrast in data lead to the hypothesis verification inconclusive. Alongside, the correlation between these two factors (i.e., total commits and corresponding ranks) provides statistically insignificant results, with *r = 0.58* with a statistical confidence rating, *p-value = 0.07 (>.05)*. Therefore, the study does not find any support from the available data to confirm this hypothesis and hereby rejected. However, having comprehensive dataset might lead to better conclusion with higher confidence.

5. FUTURE RESEARCH

Evaluation of Hypothesis H1 establishes correlation between the number of OSS projects contributed to that of the rank of contributing developers. However, related studies reported that top ranked developers are inclined towards high profile projects with long traits of development record, huge customer base, and having high customer rating and market share. Therefore, investigating the impact of these project factors in defining the ranks would increase the comprehensiveness of the reported result.

Among the other findings of this study, the positive correlation between the length of participation and the rank is established. However, related data also reveals that developers at the same rank has large variance in the number of years contributed. This observation leads to assume that there might be other associated factors that could explain this oddity. For instance, consistency of contribution measured in terms of man month can shed effective insight.

Additionally, quantity of contribution can be the other decisive factor. This study establishes a positive correlation between the quality of contribution to that of the rank of a developer. However, the quality assessment approach one of the many perspectives to be evaluated. Therefore, future study should consider factors tightly related to the quality assessment of the contributions, e.g., direct rating of the commits made by the developers, severity level of the issues / features resolved through the commits.

6. VALIDITY THREAT

Study of this nature is subject to construct validity. The core concern related to construct validity is the data source and its' authenticity. Data for this research is collected from OpenHub data source which is an established name in patronizing OSS project data by leveraging all valid project sources. Detail of OpenHub is already presented in Section 3.1. Additionally, several contemporary researches have utilized this data source, and many research sites lists this forum as a dependable source for OSS project data.

Another common concern related to construct validity is the missing data. Even though the data provided by OpenHub is highly structured and consistent, yet in few cases there are missing data. Whatever the reason might be, it is definitely detrimental towards the comprehensiveness of the research and the result reported. This research has partially suffered from this issue in case of Hypothesis H4. However, this deficiency is duly acknowledged in reporting the results in Section 4.4.

7. CONCLUSION

With commercial grade software development, Open Source Software (OSS) Projects have established a firm ground in the world software landscape for the last couple of decades. With their unorthodox development process and practices, OSS ecosystem traditionally remains the core interest of the developers from all round the globe. Thousands of qualified techno professionals join OSS projects and contribute voluntarily without monetary compensation. As such, OSS projects and people become the centre focus of many software engineering research.

One of the widely researched topics was what motivates developers to participate in OSS projects, how they get recruited and integrated in the projects. However, the other half of the story remains unexplored - *what factors related to developer's contribution lead to build a sustainable reputed career?*

This research explores the same empirically by leveraging data from OpenHub repository. The transcript of assessment shows that plurality, quality and length of service have strong positive correlation with developers' current reputation. That is, developers' reparation and acceptance within the OSS ecosystem grows proportionally with greater quality contribution in diverse OSS projects for a substantial period of time. However, amount of contributions during a developers' service life have little impact on their reputation.

REFERENCES

1. Oscar Franco-Bedoya (2015). Open Source Software Ecosystems: Towards a Modelling Framework, International Federation for Information Processing, OSS 2015, IFIP AICT 451, pp.171–179.
   https://doi.org/10.1007/978-3-319-17837-0_16
2. Jansen, S., Brinkkemper, S., Finkelstein, A.: Business network management as a survival strategy: A tale of two software ecosystems. In: Proceedings of the 1st Work- shop on Software Ecosystems, CEUR-WS, pp. 34–48 (2009).
3. Mens, T., Claes, M., Grosjean, P., Serebrenik, A.: Studying evolving software ecosystems based on ecological models, pp. 297–326. Springer, Heidelberg(2014)
   https://doi.org/10.1007/978-3-642-45398-4_10
4. Yu, E., Deng, S.: Understanding software ecosystems: A strategic modeling approach. In: Proceedings of the 3th WorkshoponSoftwareEcosystems,IWSECO,pp.65–76 (2011)
5. Yamakami, T.: A three-layer view model of oss: Toward understanding of diversity of oss. In: Proceedings of the 13th International Conference on Advanced Communication Technology, ICACT, pp. 1190–1194(2011)
6. Wynn Jr, D., Boudreau, M.C.,Watson, R.: Assessing the Health of an Open Source Ecosystem. GI Publishing, New York(2008).
7. Hoving, R., Slot, G., Jansen, S.: Python: Characteristics identification of a free open source software ecosys- tem. In: Proceedings of the 7th IEEE International Con- ference on Digital Ecosystems and Technologies, DEST, pp. 13–18(2013)
   https://doi.org/10.1109/DEST.2013.6611322

8. Elliott M, Scacchi W. 2003. Free software developers as an occupational community: Resolving conflicts and fostering collaboration. In Proceedings of the ACM Inter- national Conference on Supporting Group Work, Sanibel Island, FL, November 2003,21–30.

   https://doi.org/10.1145/958160.958164

9. Elliott M, Scacchi W. 2004. Free software development: Cooperation and conflict in A virtual organizational culture. In Free/Open Source Software Development, Koch S (ed.). Idea Group Publishing: Hershey, PA,152–172.

10. EspinosaJ.A., KrautR.E., SlaughterS.A., LerchJ.F., Herbsleb J.D., Mockus A. 2002. Shared mental models, familiarity, and coordination: A multi-method study of distributed software teams. In International Conference Information Systems, Barcelona, Spain, December, 2002425–433.

11. Jeffrey A. Roberts, Il-Horn Hann, and Sandra A. Slaughter. 2006. Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. Manage. Sci. 52, 7 (July 2006),984–999.

    https://doi.org/10.1287/mnsc.1060.0554

12. Guido Hertel, Sven Niedner, Stefanie Herrmann. 2003. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel, Research Policy,Volume32,Issue7,Pages1159-1177.

13. MahbubulSyeedM.M.,HammoudaI.andSystaT.2013. Evolution of Open Source Software Projects: A Systematic Literature Review, Journal of Software,Vol.8,No.11.

    https://doi.org/10.4304/jsw.8.11.2815-2829

14. Scacchi W., Feller J., Fitzgerald B., Hissam S. and Lakhani K., Understanding Free/Open Source Software Development Processes, Software Process Improvement and Practices,2006.

15. Ghosh R, Prakash VV. 2000. The orbiten free software survey. First Monday 5(7): Also see http://www.infonomics.nl/FLOSS/ • for further information.

16. HannI-H,RobertsJ,SlaughterS,FieldingR.2002.Econo mic incentives for participating in open source software projects. In•Proceedings of the Twenty-Third Inter- national Conference on Information Systems, December 2002365–372.

17. Hars A, Ou S. 2002. Working for free? Motivations for participating in open source projects. International Jour- nal of Electronic Commerce 6(3):25–39.

18. Madey G, Freeh V, Tynan R. 2004. Modeling the F/OSS community: A quantative investigation. In Free/Open Source Software Development, KochS(ed.).Idea Group Publishing: Hershey, PA,203–221.

19. Hertel G, Neidner S, Hermann S. 2003. Motivation of software developers in open source projects: an internet based survey of contributors to the Linux kernel. Re- search Policy 32(7):1159–1177.

    https://doi.org/10.1016/S0048-7333(03)00047-7

20. Nakakoji K, YamamotoY, NishinakaY, Kishida K,Ye Y. Evolution patterns of open-source software sys tems and communities. International Workshop Principles of Software Evolution,76–85.

21. Huntley CL. 2003. Organizational learning in open-source software projects: An analysis of debugging data. IEEE Transactions on Engineering Management 50(4):485–493.

22. LernerJ,TiroleJ.2002.Some simple economics of open source. Journal of Industrial Economics 50(2):197–234.

    https://doi.org/10.1111/1467-6451.00174

23. Stewart KJ,Gosain S.2001.An exploratory study of ideology and trust in open source development groups. In Proceedings of the 22nd International Conference Information Systems, New Orleans,LA.

24. Gacek C, Arief B. 2004. The many meanings of open source. IEEE Software 21(1):34–40.

25. Black Duck Open Hub, https://www.openhub.net/,last visited on May, 2020.

26. Black Duck software, https://github.com/blackducksoftware/ohlohapi, last visited on May, 2020.

27. Robin Luckey, ”Ohloh Kudos”. OHLOH. May, 2007 (Retrieved 29 February2008).

28. Bill Snyder,”Is counting open-sourcecode contributions really useful?”. InfoWorld. February 2008 (Retrieved 29 February2008).

29. Snapshot of Data Used, https://drive.google.com/drive/ folders/1VJivG0g NxjDY8oQJ98iEhrr0D-7mVDAZ? usp=sharing

30. YunwenYeandKouichiKishida.2003.Towardanunderstanding of the motivation Open Source Software developers. In Proceedings of the 25th International Conference on Software Engineering (ICSE ’03). IEEE Computer Society, USA,419–429.

    https://doi.org/10.1109/ICSE.2003.1201220

31. Raymond, E.S., and B. Young. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O’Reilly, 2001.

32. Lave, J., and E. Wenger. Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, Cambridge, UK,1991.

33. Maturana,H.R.,and F.J.Varela.The Tree of Knowledge: The Biological Roots of Human Understanding. Shambhala Publications,1998.

34. Singh, B., Gautam, S. (2016). The Impact of Software Development Process on Software Quality. International Conference on Computational Intelligence and Communication Networks (CICN); vol.10,pp:09-10

35. R. Breu, A. Kuntzmann-Combelles and M. Felderer (2014) ”New Perspectives on Software Quality [Guest editors’ introduction],” in IEEE Software, vol. 31, no. 1, pp.32-38

    https://doi.org/10.1109/MS.2014.9.

36. M. M. Mahbubul Syeed, Alexander Lokhman, Tommi Mikkonen, and Imed Hammouda. Pluggable Systems as Architectural Pattern: An Ecosystemability Perspective. In Proceedings of the 2015European Conference on Software Architecture Workshops. ACM, Article 42, 1–6. 2015.
37. Rohini B. Jadhav,Shashank D. Joshi,Umesh G.Thorat,Aditi S. Joshi, A Software Defect Learning and Analysis Utilizing Regression Method for Quality Software Development. International Journal of Advanced Trends in Computer Science and Engineering. 8 (4). pp. 1275-1282. 2019. https://doi.org/10.30534/ijatcse/2019/38842019
38. K Chaitanya1, Dr. K Rajasekhra Rao2, Dr. JKR Sastry, A Framework for Testing Distributed Embedded Systems. International Journal of Advanced Trends in Computer Science and Engineering. 8 (4). pp. 1194-1227. 2019. https://doi.org/10.30534/ijatcse/2019/30842019
39. M. M. Mahbubul Syeed, Imed Hammouda. Socio-Technical Dependencies in Forked OSS Projects: Evidence from the BSD Family. Journal of Software, 9(11),pp 2895-2909, 2014.