# Cave or Community?
# An Empirical Examination of 100 Mature Open Source Projects [1]

Sandeep Krishnamurthy
Assistant Professor of E-Commerce and Marketing
Business Administration Program
University of Washington, Bothell
18115 Campus Way NE, Room UW1-233
Bothell, WA 98011-8246

Tel: (425) 352-5229
Fax: (425) 352-5277
E-Mail: sandeep@u.washington.edu
Web URL: http://faculty.washington.edu/sandeep

May 2002

[1]The author thanks Julie Thornton of Linux International for invigorating conversations that informed his thinking. The Open Source list run by Karim Lakhani of MIT(opensource.mit.edu) has been an awesome source of inspiration for this work.

# Cave or Community?
# An Empirical Examination of 100 Mature Open Source Projects


## ABSTRACT

Starting with Eric Raymond's groundbreaking work, The Cathedral and the Bazaar, open-source software (OSS) has commonly been regarded as work produced by a community of developers. Yet, given the nature of software programs, one also hears of developers with no lives that work very hard to achieve great product results. In this paper, I sought empirical evidence that would help us understand which is more common- the cave (i.e., lone producer) or the community. Based on a study of the top 100 mature products on Sourceforge, I find a few surprising things. First, most OSS programs are developed by individuals, rather than communities. The median number of developers in the 100 projects I looked at was 4 and the mode was 1- numbers much lower than previous ones reported for highly successful projects! Second, most OSS programs do not generate a lot of discussion. Third, products with more developers tend to be viewed and downloaded more often. Fourth, the number of developers associated with a project is unrelated to the age of the project. Fifth, the larger the project, the smaller the percent of project administrators.

## Introduction

Starting with Eric Raymond's ground-breaking work, The Cathedral and the Bazaar, open-source software (OSS) has commonly been regarded as work produced by a community of developers. Ghosh's cooking pot markets, similarly, point to a communal product development system. Certainly, this is a good label for some OSS products that have been featured prominently in the news. For instance, Sproull and Moon point out that by July 2000, about 350 contributors to LINUX were acknowledged in a credits list in the source code of the kernel.

However, my goal in this paper is to ask if the community-based model of product development holds as a general descriptor of the average OSS product. I systematically look at the actual number of developers involved in the production of one hundred mature OSS products. What I found is more consistent with the *lone developer (or cave) model of production* rather than a community model (with a few glaring exceptions, of course).

This is not to say that there is no community in the OSS movement. For instance, the findings of Butler, Kiesler, Sproull and Kraut (2002) point to participation by individuals other than the creators of OSS-program-related mailing lists. My contention is only that communities do things other than produce the actual product- e.g. provide feature suggestions, try products out as lead users, answer questions etc. Formally separating software production from other steps in the development of OSS programs will provide greater clarity to the discussion of the OSS phenomenon.

## Methodology

As many in this audience will be aware, Sourceforge.net is a large repository of OSS programs. Sourceforge.net places OSS programs into six categories based on their stage of product development- Planning, Pre-Alpha, Alpha, Beta, Production/Stable and Mature. As of May 2, 2002, the number of projects in each stage was as given below-

1 - Planning (8262 projects)
2 - Pre-Alpha (5533 projects)
3 - Alpha (4907 projects)
4 - Beta (5727 projects)
5 - Production/Stable (4365 projects)
6 - Mature (480 projects)

It is fair to say that only a small percent of all programs make it to the Mature stage(i.e., category 6). Therefore, choosing products in this category allows us to focus on the products with the best chance to build a community around them. Products in the early stages of development may be small and not

require a lot of assistance. It also takes time to build a community around a product. Mature products that have been out for a while(on average, the projects studied here were founded in October 2000- most had made several product releases) have had more time to build a community.

To be more specific the top 100 most active projects (based on Sourceforge's activity percentile) in the mature class were chosen for this study. This represented about 20% of all mature programs. A dataset of the characteristics of these programs was manually compiled and is attached as an Appendix[1]. Data was collected from 23rd April to May 1st of 2002.

Findings

For the findings reported here, the OSS program was the unit of analysis. Our findings are limited to the 100 projects studied here. No claims are made for generalizing these findings to the universe of OSS projects. We leave that to future research.

The first main finding was that-

*Finding 1: The vast majority of mature OSS programs are developed by a small number of individuals.*

This was the most surprising finding of all. As shown in Table 1, the median number of developers involved in the 100 projects studied here was 4 and the mode was 1. Sourceforge allows the designation of some developers as project administrators. The median number of project administrators was 1. In fact, the largest number of developers in a project was 42- a far cry from the high numbers reported previously. It is also important to note that there was great variation in the number of developers among these programs- the standard deviation was 8.24.

**Table 1**
**Descriptive Statistics of Developers and Project Administrators**

| | Number of Project Administrators | Number of Developers |
|---|---|---|
| Mean | 2.21 | 6.61 |
| Median | 1 | 4 |
| Mode | 1 | 1 |
| Minimum | 1 | 1 |
| Maximum | 14 | 42 |
| Std. Deviation | 1.91 | 8.24 |

Moreover, as shown in Table 2, only 29% of all projects had more than 5 developers while 51% of projects had 1 project administrator. Only 19 out of 100 projects had more than 10 developers. On the other extreme, 22% of projects had only one developer associated with them.

**Table 2**
**Frequency Distribution of the Number of Project Administrators and Developers**

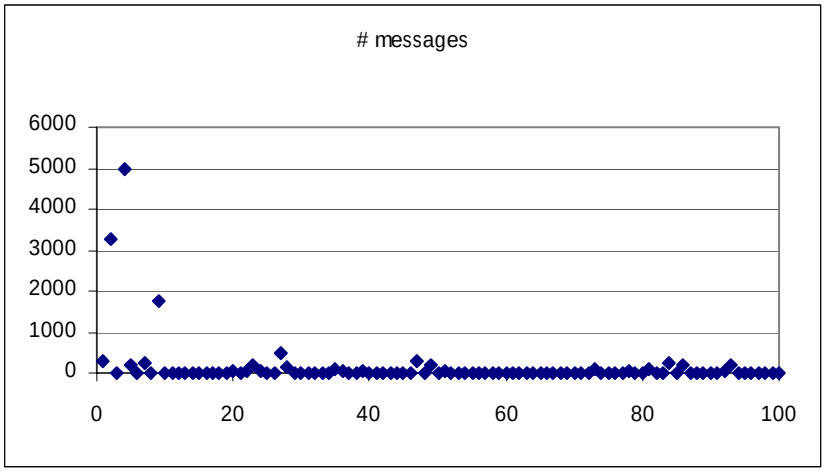
| | Project Administrators | Developers |
|---|---|---|
| 1 | 51 | 22 |
| 2 | 22 | 12 |
| 3 | 6 | 10 |
| 4 | 11 | 12 |
| 5 | 6 | 15 |
| >5 | 4 | 29 |

*Finding 2: Very few OSS products generate a lot of discussion. Most products do not generate too much discussion.*

On average, each OSS product had 2 forums and 2 mailing lists for discussions pertaining to the product. Ten of the 100 products had neither an online forum nor a mailing list, 21 products did not have a mailing list associated with them and 33 products did not have an online forum associated with them.

The total number of messages in the forums assigned for discussion of these products is shown in Figure 1. The vast majority of them led to very few messages over the life time of the product. In fact, 33 out of 100 projects had 0 messages! At the same time, a few products led to great discussion with the highest number of messages over a life time of a product standing at 4,952.

**Figure 1**
**Number of Messages in Official Forums Over the Life Time of an OSS Product**



*Finding 3:* *Products with more developers tend to be viewed and downloaded more often.*

Figures 2 and 3 clearly show the trends. The page views and downloads are over the life time of the project. The actual correlation between the number of developers and page views is 0.56 and that between the number of developers and downloads is 0.27.

**Figure 2**
**Page Views vs. Number of Developers**

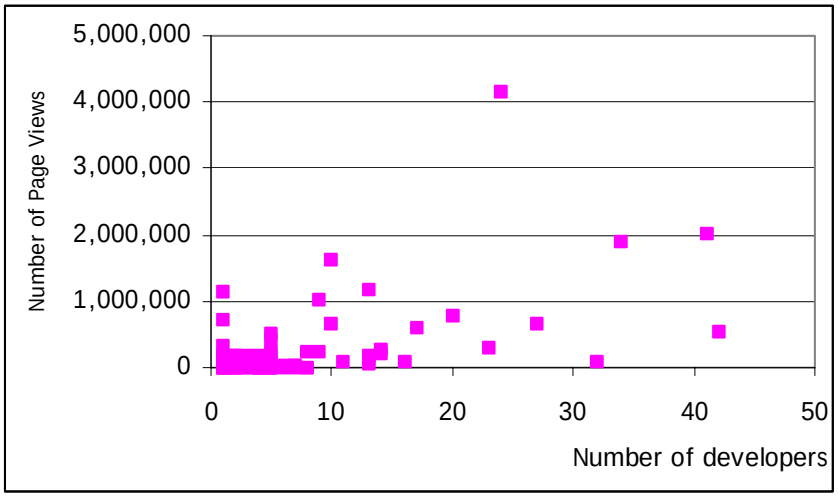
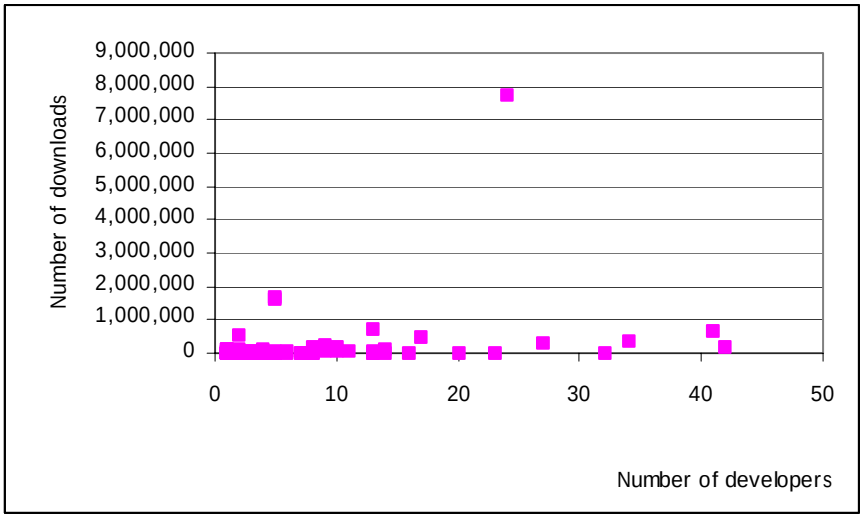


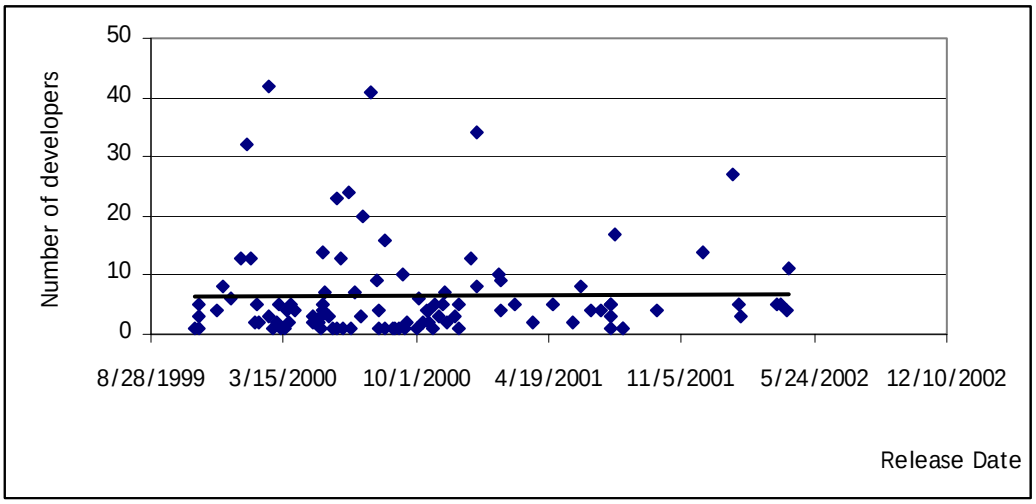

**Figure 3**
**Number of downloads vs. Number of developers**



*Finding 4: The number of developers working on a OSS program was unrelated to the release date.*

It could be argued that older projects may have more developers associated with them. However, we found no relationship between the release date and the number of developers associated with a program. Figure 4 makes this clear.

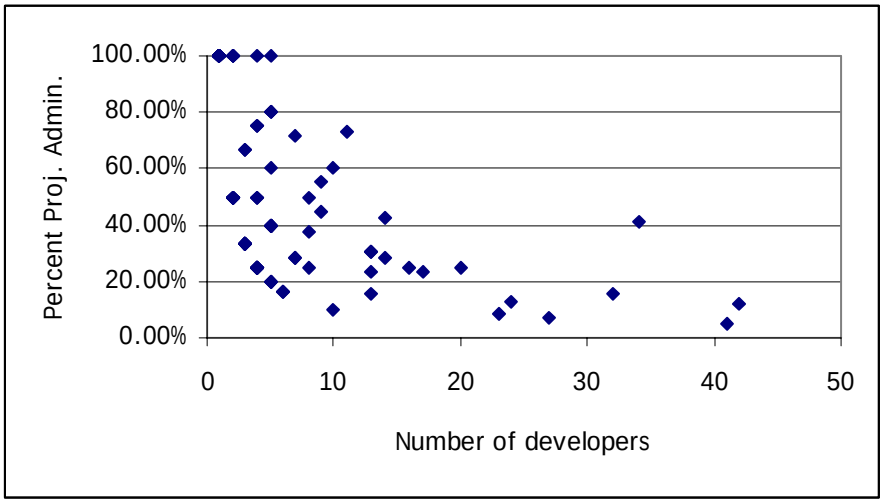
**Figure 4**
**Release Date vs. Number of Developers**




*Finding 5: A smaller percent of participants were assigned as project administrators in larger groups.*

This is to be expected. The trend is shown in Figure 4 below.

**Figure 5**
**Percent of Project Administrators vs. Total Number of Developers**



Discussion of Findings

The findings in this study are actually consistent with many previous papers on OSS products. For instance, an analysis of top 100 most prolific contributors identified by the 2000 Orbiten Survey is shown in Table 3. Of the top 100, 70 were individuals or very small groups(typically pairs). These individuals accounted for 46.1% of the code and 50.4% of projects. One individual had contributed to 267 projects.

**Table 3**
**Who produces OSS programs?**
**An analysis of the top 100 most prolific contributors.**
(Source: The Orbiten Free Software Survey
1st Edition, May 2000, http://orbiten.org/ofss/codd-render.cgi?action=project&sortkey=projects )

| Category | Number of programs | Bytes | Percent of Top 100 Total | Number of Projects | Percent of Top 100 Total | Most Projects by Participant in Category |
|---|---|---|---|---|---|---|
| For-profit organization | 14 | 56,493,879 | 13.3% | 193 | 10.2% | 66 |
| Non-profit org/community | 4 | 132,347,379 | 31.0% | 586 | 31.1% | 546 |
| University | 4 | 20,392,109 | 4.8% | 156 | 8.3% | 156 |
| Individuals/small groups | 70 | 196,738,432 | 46.1% | 951 | 50.4% | 267 |
| Author unknown | 8 | 20,335,032 | 4.8% | N/A | N/A | N/A |
| TOTAL of top 100 | 100 | 426,306,831 | 100% | 1,886 | 100% | N/A |

Similarly, previous authors have identified the strong hand of the leader of an OSS program. Sproull and Moon refer to Linus Torvalds as a “great man”. Others have pointed out that Torvalds essentially did not have a life and spent considerable number of hours rewriting code submissions by others.

Even though the discussion here may seem like an example of extreme free-riding, the reader needs to know that all free-riding is not necessarily "bad". For instance, consider public radio stations in the United States. Even the most successful stations have about a 10% contribution rate or a 90% free-ridership rate. But, they are still able to meet their goals! Similarly, the literature on lurking in e-mail lists has suggested that if everyone in a community contributes it may actually be counter-productive.

Similarly, a recent survey of participants in open-source projects conducted by the Boston Consulting Group and MIT provides more insight. The top five motivations of open-source participants were-

1. To take part in an intellectually stimulating project.
2. To improve their skill.
3. To take the opportunity to work with open-source code.
4. Non-work functionality.
5. Work-related functionality.

Interestingly, motivations such as defeating proprietary software ranked low. This paints a picture of the motivated developer who wants to create a product that is interesting. This is consistent with our findings.

Learner and Tirole have proposed that attracting developers is a difficult task. "Open source developers work on projects that they consider important and significant additions to the software universe. They are not interested in products that would lead to a dead end or would make a small and marginal impact." Perhaps, what we are watching is the process by which smaller projects get turned down. The strong characteristics of a few projects is strongly reminiscent of the winner-take-all structure proposed by Adar and Huberman.

Obviously, this study has its own limitations. One could argue that projects in other categories may not have similar characteristics. Preliminary results indicate that open source projects in other categories may also exhibit similar properties. Table 3 summarizes the descriptive statistics for the top 20 projects in all other stages. Group sizes are, in general, much higher than for mature projects. However, they are still low- the median ranges from 6.5 to 8- and much lower than what some may perceive. Future research must conduct a larger comparison.

**Table 3**
**Descriptive Stats for Top 20 Projects Across Stages**

| | Planning Stage | Pre-Alpha | Alpha | Beta | Production/ Stable |
|---|---|---|---|---|---|
| mean | 8.7 | 16.6 | 16.8 | 12.1 | 12.8 |
| median | 6.5 | 6.5 | 8.0 | 8.5 | 8.0 |
| min | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| max | 25.0 | 99.0 | 129.0 | 61.0 | 54.0 |
| std. Dev. | 6.3 | 23.6 | 28.0 | 13.2 | 14.1 |

## Conclusion

As an academic community, it is important that we distinguish between producers of OSS programs and others. The community model is a poor fit for the actual production of the software. While some products that are very well publicized may attract large number of developers, most OSS products are developed and maintained by a tiny number of developers. In many cases, these products are not even discussed or talked about.

Perhaps, there is some merit to clearly delineating the relative roles of individuals, communities and social networks. Some have already proposed moving away from the term community to the term voluntary association. This study may help in that discussion.

## Endnotes

[1] The author is grateful to his student, Lisa Kim, for assisting in data collection. The Appendix is omitted here for brevity. If you are interested in taking a look, e-mail me at sandeep@u.washington.edu.