An Empirically-Based Criterion for Determining the Success of an Open-Source Project

Dror G. Feitelson
School of Computer Science & Engineering
The Hebrew University
91904 Jerusalem, Israel
E-mail: feit@cs.huji.ac.il

Gillian Z. Heller
Department of Statistics
Macquarie University
Sydney, NSW 2109, Australia
E-mail: gheller@efs.mq.edu.au

Stephen R. Schach*
Dept. Electrical Engineering & Computer Science
Vanderbilt University
Nashville, TN 37235, USA
E-mail: srs@vuse.vanderbilt.edu

Abstract

In order to determine a success criterion for open-source software projects, we analyzed 122,205 projects in the SourceForge database. There were 80,597 projects with no downloads at all. We restricted our analysis to the 41,608 projects that together were downloaded 704,897,520 times. Contrary to what we had expected, the distribution of the number of downloads of each project is not Zipf-like; only a portion of the log-log plot of the number of downloads and their rank appears to be a straight line. We performed least-squares analysis (utilizing the Bayesian information criterion) to divide the plot into three segments. On the basis of the shapes of the corresponding curves and the locations of their boundary points, we categorized the projects as follows: 85 superprojects (highly successful projects with more than 1.1 million downloads); just over 10,000 successful projects (with more than 1680 downloads each); and struggling projects (with 1680 downloads or fewer). In terms of our criterion, only a quarter of the projects that have one or more downloads can be deemed to be successful.

Keywords: *open-source software; distribution of downloads; Zipf-like distribution; SourceForge repository; success criterion.*

"A certain number ... must I select from all"

William Shakespeare,
The Tragedy of Coriolanus, Act I, Scene VI.

*Corresponding author

1 Introduction

We are currently investigating the factors that lead to a successful open-source project. The first step is to determine a criterion for a successful open-source project that is directly derived from empirical data. That criterion is the subject of this paper.

The obvious measure of success is the number of times that a project is downloaded. That is, if the number of downloads exceeds some threshold value to be determined, the project is deemed to be a successful one. Our contribution is to use a discontinuity in the distribution of the number of downloads to determine the threshold value.

The SourceForge web site provides storage and support for open-source[1] projects. Open-source software can be developed at and downloaded from other sites, such as FreshMeat, but SourceForge is by far the largest site of this kind. At the time that we collected our data (May 2005), 122,205 projects were listed there. Of these, 41,608 were downloaded at least once. In addition, the number of members exceeded one million. However, one does not have to be a member to download software from SourceForge, so the actual number of users is surely much greater than this figure. The large number of projects and high volume of activity at the SourceForge web site provide us with sufficient data to make observations regarding the distribution of projects.

The remainder of this paper is organized as follows: In Section 2, we describe the SourceForge data. Criteria for

[1] In this paper, we use the term "open source" to denote both open-source software [www.opensource.org/docs/definition.php] and free software [www.gnu.org/philosophy/free-sw.html]

successful open-source projects are outlined in Section 3. Zipf's Law is the subject of Section 4. Our results regarding the distribution of downloads appear in Section 5. In Section 6, we consider successful projects. Our conclusions appear in Section 7.

2 SourceForge Data

We analyzed a snapshot of the SourceForge database as of May 2005 [9]. At that time, a total of 122,205 projects had been registered with SourceForge.

Anyone can register a project at sourceforge.net. In fact, we have even seen courses on software production where course projects were done on SourceForge (e.g., projects musearchengine, group5, and loanarranger). In some cases, such projects later seem to have become "real" projects (e.g. project travelsearcheng). In addition, many of the projects stored on SourceForge are placeholders for projects that have yet to get off the ground. Accordingly, it is not surprising that the database reflected that 80,597 of the 122,205 projects (66 percent) have never been downloaded.

Most of these 80,597 projects with no downloads have only one developer. However, there are a number of these projects with more than one developer. We conjecture that at least some of these projects use SourceForge for coordination and to list the developers, but use some other channel for downloads.

We decided to err on the side of safety. Accordingly, in this paper we have ignored the 80,597 projects with no downloads, concentrating instead on the 41,608 projects that together were downloaded a total of 704,897,520 times.

3 What Constitutes a Successful Open-Source Project?

Some of the most successful software products of all time are open-source projects, including Linux (operating system), Apache (Web server), MySQL (database management system), PHP (scripting language), and Firefox (Web browser). Millions of copies of each of these products have been downloaded. Conversely, as stated above, numerous open-source projects have failed.

An obvious question is: What constitutes a successful open-source project? One important criterion is commercial acceptance, that is, widespread use as mission-critical software by a large number of companies [12]. The problem with this criterion is that the data needed to determine whether this criterion is satisfied are generally not available; few companies are willing to divulge information about their mission-critical software.

Instead, we need to be able to decide whether a project is successful in terms of measurable quantities. One such quantity that is available on the SourceForge database is the number of downloads.

There is no question that a project that has been downloaded hundreds of thousands of times is successful. But what about a project that has been downloaded only 75 times? At first sight, what is needed is a criterion such as the following:

> "A project is successful if it has been downloaded over 250 times"

There is a major problem with criteria of this type: The parameter (250 here) is totally arbitrary; other researchers can come up with a different parameter that appears to be equally reasonable. Instead, what is needed is an empirically-based criterion, that is, a way of characterizing a successful open-source project based on data such as the SourceForge database.

4 Zipf's Law

Success is related to popularity. We therefore start with measures of popularity in other areas.

In a large corpus of text, there will be a few words such as "the" and "and" (or their equivalent in a natural language other than English) that occur frequently, but many words that occur rarely. George Kingsley Zipf (1902-1950), a Harvard linguistics professor, studied the frequencies of the use of words in English text. He tabulated the number of times n that each word appeared in a specific text. He then sorted the words in rank order, with the most popular first. He found that the number of occurrences n is related to the rank r according to [13]

$$n \propto \frac{1}{r} \tag{1}$$

That is, *Zipf's Law* states that the frequency of use of the rth-most-frequently-used word in any natural language is approximately inversely proportional to r.

The *Zipf distribution*, namely, $n \propto r^{-1}$, has been generalized; a *Zipf-like* distribution[2] is one for which

$$n \propto r^{-a} \tag{2}$$

(For the Zipf distribution, the value of a is equal to 1.) A simple way to visualize this power law is to take the logarithm of both sides of Equation (2), yielding the equation

$$\log n \propto -a \log r \tag{3}$$

Accordingly, we plot $\log n$ against $\log r$. If the points lie on a straight line then we conclude that the data are distributed according to the Zipf-like distribution. In addition, if the value of the slope a is 1, then we have a Zipf distribution.

[2] Adamic has shown that the Zipf-like distribution is equivalent to the Pareto distribution [1].

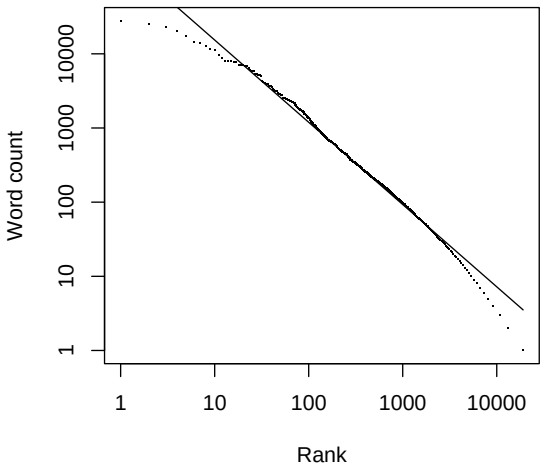

Figure 1. *Log-log distribution of the frequencies of the words in Shakespeare's canon.*

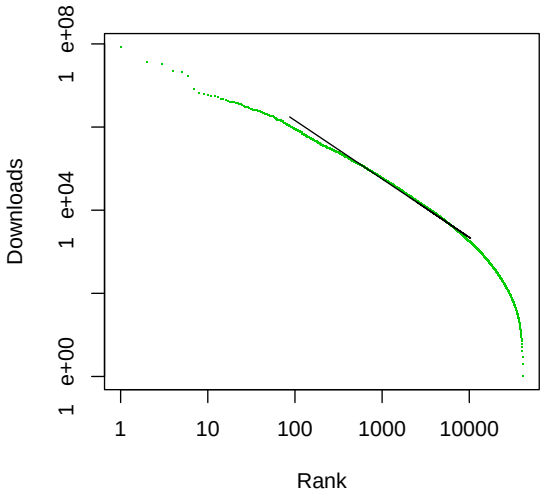

Figure 2. *Log-log plot of number of downloads.*

For example, consider the frequencies of the 915,511 words that make up the canon of Shakespeare's plays [ise.uvic.ca/Annex/Stats/FreqAscending.html]. Figure 1 shows that the log-log plot of the frequencies and ranks of the 23,321 distinct words in this corpus is approximately a straight line[3].

Zipf and Zipf-like distributions are encountered in many other fields. For example, city sizes follow a Zipf distribution. These distributions also occur all over the Internet. For example, Adamic and Huberman [2] found that the number of pages in a website, the number of visitors to a website, the number of links to other sites, and the number of links from other sites are all distributed according to the Zipf-like distribution[4]. They also found that the connectivity of the internet backbone at the autonomous system level fits the Zipf-like distribution. They even found that the number of unique AOL visitors to websites on December 1, 1997 obeys Zipf's Law. Other researchers have found other instances of the Zipf-like distribution on the Internet. For example, Fall found that packets in a user's telnet session and bytes in ftp data are distributed according to the Zipf-like distribution [5]. The size of files on a Calgary University web server also has this distribution [4], as does the number of distinct local university hosts to which different remote KaZaA hosts are connected [11]. Hunt and Johnson showed that the distribution of downloads from SourceForge over a 30-day period is approximately Zipf-like [8]. In short, the Zipf-like distribution (and its special case, the Zipf distribution) permeates the Internet.

Huberman and Adamic [7] have given a theoretical explanation as to why the Zipf-like distribution is so prevalent on the Internet. Their generative growth model is outlined in Section 6.

5 Distribution of Downloads

At the SourceForge site, the projects are ranked according to the number of downloads. In fact, the number of downloads is quite widely regarded as the most direct measure of popularity and, hence, success. It is therefore natural to expect the distribution of downloads to be Zipf-like. Accordingly, we determined the number of downloads for each SourceForge project, and ranked these numbers. We plotted the logarithm of the number of downloads against the logarithm of the project rank on the basis of the number of downloads. Surprisingly, as shown in Figure 2, only the points in the center of the plot appear to lie on a straight line.

In more detail, the data points in Figure 2 appear to fall into three groups:

[3]In Figures 1 and 2, multiple instances have been ignored when computing a straight-line fit. For example, in Figure 1, the point that denotes the 8,641 words that occur only once (like "mop" or "honorificabilitudinitatibus") is given the same weight as the point that denotes the 33 words (like "number") that occur 59 times and the point denoting the single word ("the") that occurs 27,921 times in the Shakespearean canon. Otherwise, the thousands of words that occur only once would overwhelm the handful of common words that occur tens of thousands of times.

[4]The authors of the papers cited in this and the following paragraph used the term "Pareto distribution" rather than "Zipf-like distribution." As explained in Footnote 2, the two terms are synonymous.

1. The projects on the left side of the diagram are outliers. We term them superprojects. After all, there can be no doubt that projects with over a million downloads are successful.

2. The central part of the diagram appears to be a straight line. That is, this portion of the plot has a Zipf-like distribution.

3. The right part of the diagram, corresponding to the vast majority of the 41,608 projects with at least one download, curves sharply downward. That is, there are fewer projects that have a small number of downloads than would be expected if all the downloads were Zipf-like.

In order to determine the boundaries of the central (straight-line) portion of the curve of Figure 2, we specified the following mathematical model for the log-log relationship:

$$\hat{y}_i = \begin{cases} \alpha_1 + \beta_1 x_i + \gamma_1 x_i^2 + \delta_1 x_i^3 & x_i < c_1 \\ \alpha_2 + \beta_2 x_i & c_1 \le x_i \le c_2 \\ \alpha_3 + \beta_3 x_i + \gamma_3 x_i^2 + \delta_3 x_i^3 & x_i > c_2 \end{cases}$$

where $y_i = \log(\text{downloads})$ and $x_i = \log(\text{rank})$ of the ith project, and $\hat{y}_i$ is the model estimate of y_i. To choose the optimal values of the boundaries c_1 and c_2, we used the least-squares criterion, that is, we chose those values of c_1 and c_2 and the parameters $\alpha_1, \ldots, \delta_3$ that minimized

$$SSE = SSE_1 + SSE_2 + SSE_3$$

where

$$SSE_k = \sum_{i=1}^{n_k} \left[y_i - \left(\alpha_k + \beta_k x_i + \gamma_k x_i^2 + \delta_k x_i^3 \right) \right]^2 \qquad k = 1, 3$$

$$SSE_2 = \sum_{i=1}^{n_2} \left[y_i - (\alpha_2 + \beta_2 x_i) \right]^2$$

and n_k is the number of observations (projects) falling into segment k, $k = 1, 2, 3$. (The observations in segment k have been renumbered as $1, \ldots, n_k$.)

We needed, however, to refine this least-squares criterion because, as it stands, the optimal solution will always be $c_1 = c_2$, that is, with no linear middle segment. This happens because a model with more parameters (such as the cubic models in segments 1 and 3) will always produce a better fit, and therefore a lower sum of squares, than a model with fewer parameters (such as the linear segment). We therefore need to apply a penalty to observations falling in the outer segments. We used the Bayesian information criterion (BIC), which is used to trade off goodness of fit and parameter precision in statistical model selection [10]. The term $p_k \log(n_k)$ is added to each SSE_k, where p_k is the number of parameters in segment k, that is, $p_1 = p_3 = 4$ and $p_2 = 2$. This penalizes segments 1 and 3 more heavily than segment 2. The criterion which we minimized is then

$$\sum_{k=1}^{3} (SSE_k + p_k \log(n_k))$$

The optimal solution is given by $c_1 = 4.45$ and $c_2 = 9.25$, translating to a linear segment between 86 and 10,358 projects[5].

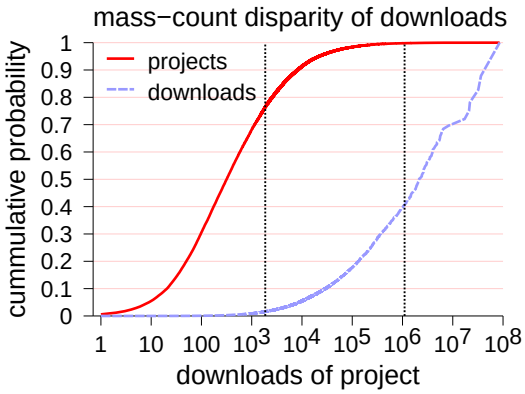

Figure 3. *Demonstration of mass-count disparity in downloads.*

Next, we analyzed the 41,608 projects with at least one download each from the viewpoint of mass-count disparity [6], a measure of the disparity between the distribution of mass (in our case, downloads) and the distribution of count (in our case, projects) in a heavy-tailed distribution. The results are shown as reflected in Figure 3.

The top line shown in this figure is the distribution of projects with different numbers of downloads. The bottom line is the "mass distribution" [3]:

$$F_m(x) = \frac{\int_{-\infty}^{x} x f(x) dx}{\int_{-\infty}^{\infty} x f(x) dx}$$

where $f(x)$ is the pdf of the projects distribution. This represents the probability that a download belongs to a project with fewer than x downloads. The disparity between the two lines indicates that a few very popular projects are the targets of most downloads, whereas most projects receive only a few downloads — reminiscent of Pareto's Law[6].

[5] Mathematically, the cutoff should be at 10,353 projects. However, the projects with ranks 10,353 through 10,358 all have 1,681 downloads. In view of the fact that these six projects are all represented by the same point in Figure 2 (see Footnote 3), the cutoff is actually at rank 10,358.

[6] Vilfredo Pareto (1848-1923) was a great Italian economist and sociol-

	Range of ranks on the basis of the number of downloads	Percentage of projects with at least one download	Range of number of downloads	Percentage of all downloads
Super	1 to 85	0.2%	85,463,102 down to 1,111,396	59.4%
Successful	86 to 10,358	24.7%	1,079,981 down to 1,681	39.1%
Struggling	10,359 to 41,608	75.1%	1,680 down to 1	1.5%

Table 1. *Categorization of projects with at least one download.*

6 Success and the Distribution of Downloads

We now return to Figure 2. As previously indicated, the point at which the line representing the number of downloads starts to curve is at rank of 10,359, corresponding to 1,680 downloads. This is a major deviation from a Zipf-like distribution. Coupled with the data in Figure 3, this indicates that the downloads fall into three groups: the handful of superprojects (the first 0.2 percent of projects, but accounting for nearly 60 percent of the downloads), the next 24.7 percent of the projects (acounting for about 39 percent of the downloads), and the remaining 75.1 percent of the projects (that receive only 1.5 percent of the downloads). Our interpretation of these results is that the top 24.9 percent of the 41,608 projects (that is, the superprojects together with those lying on the straight line) may be deemed successful. One reason for this claim is that, as previously stated, the number of downloads is widely accepted as a measure of success, that is, Figure 2 may be interpreted as a graph of success (or lack thereof). A second reason is the results reflected in Figure 3. The dichotomy between the first 24.9 percent and the rest of the projects can therefore, in our opinion, be used as a way to distinguish between successful projects and those we can classify as "struggling." A third reason is now given.

Huberman and Adamic [7] have published a generative growth model that explains the prevalence of the Zipf distribution on the Internet. The basic assumption underlying their model is *proportional growth* (or *preferential attachment*). For example, the number of pages added to or removed from a site is proportional to the number of pages already present. Huberman and Adamic showed that the assumption of proportional growth leads to a log-normal distribution of the number of pages at a site. In addition, during the first ten years of the existence of the World Wide Web, new sites appeared at an exponential rate. An exponentially-weighted mixture of log-normals then yields the Zipf-like distribution [7]. Similar arguments lead to the same result for the number of downloads of open-source projects.

In other words, the prevalence of Zipf's Law on the Internet is a consequence of generative growth dynamics in which the successful gain more. Conversely, deviation from the straight line of a log-log distribution indicates a lack of success. That is, the projects on the right-hand side of Figure 2 are those that have not yet succeeded in obtaining the critical mass[7] of downloads needed to generate a following that will lead to additional downloads at a self-sustaining rate. That is why we feel that we are justified in referred to them as "struggling," even if some of them have been downloaded over 1,600 times.

We note that similar observations were made by Hunt and Johnson [8] in examining the number of downloads of projects from SourceForge over two consecutive 30-day periods in 2002. They also observed a Zipf-like distribution, but with "fewer than expected projects with a low number of downloads."

7 Results, Conclusions, and Future Work

We have examined data relating to the 122,205 open-source projects in the SourceForge database as of May 2005, focusing on the 41,608 projects that had been downloaded one or more times. The distribution of the number of downloads from each project does not appear to be Zipf-like; only the central portion of the log-log plot of the number of downloads (Figure 2) appears to be a straight line. By applying least-squares optimization to a mathematical model that incorporates the Bayesian information criterion [10], we were able to divide the projects into three categories:

Category 1: *Superprojects*. These 85 outlier projects on the left side of the plot were downloaded more than 1.1 million times each. The most popular project, emule, a peer-to-peer file-sharing client, was downloaded 85,463,102 times, and the project ranked 85th, gnu-darwin, was downloaded 1,111,396 times. (The gnu-darwin site offers a variety of free software, including the Darwin and GNU operating systems.)

ogist. He found that 80 percent of the land in Italy, at that time, was owned by only 20 percent of the population. He later discovered so many other instances of 80-20 that the ratio was later named "Pareto's Law."

[7] We have used the terms "mass-count disparity" and "critical mass" consistently in this paper. This is, the word "mass" refers to the number of downloads in both cases.

Category 2: *Successful projects.* The projects with ranks from 86 to 10,358 lie approximately on a straight line in the center of the log-log plot of the number of downloads. The number of downloads varied from 1,079,981 down to 1,681.

Category 3: *Struggling projects.* There are 41,608 projects in the SourceForge database that have been downloaded at least once. The vast majority of them (over 75 percent) have been downloaded too few times to be considered successful. These are the projects lying on the downward-sloping right side of the plot. In the light of the generative growth model of Huberman and Adamic [7], we feel that it is appropriate to consider these projects as "struggling".

Our results are summarized in Table 1.

We turn now to future work. The data we have analyzed are cumulative. That is, they reflect all downloads of each project, from its inception until the end of May 2005. Accordingly, in order to be able to test our approach on new data, we will have to wait until (say) May 2006 for a sufficient number of new data points to have accumulated. In particular, we are interested in the movement of projects between categories, and whether this correlates with what we would intuitively call "success."

References

[1] L. A. Adamic, "*Zipf, power-laws, and Pareto – a ranking tutorial*". 2000. http://www.hpl.hp.com/research/idl/papers/ranking/.

[2] L. A. Adamic and B. A. Huberman, "*Zipf's law and the Internet*". *Glottometrics* **3**, pp. 143–150, 2002.

[3] M. E. Crovella, "*Performance evaluation with heavy tailed distributions*". In *Job Scheduling Strategies for Parallel Processing*, pp. 1–10, Springer Verlag, 2001. Lect. Notes Comput. Sci. vol. 2221.

[4] A. B. Downey, "*Lognormal and Pareto distributions in the Internet*". *Comput. Commun.* **28(7)**, pp. 790–801, May 2005.

[5] K. Fall, "*Simulating the Internet: challenges and methods*". In *IX Computer Science Symp.*, Mayab University, Yucatan, Mexico, Apr 1998.

[6] D. G. Feitelson, *Metrics for Mass-Count Disparity*. Technical Report, Hebrew University, 2005.

[7] B. A. Huberman and L. A. Adamic, "*Growth dynamics of the world-wide web*". *Nature* **401**, p. 131, Sep 1999.

[8] F. Hunt and P. Johnson, "*On the Pareto distribution of Sourceforge projects*". In *Open Source Software Development Workshop*, University of Newcastle, Feb 2002.

[9] G. Madey, "*Sourceforge.net research data archive*". URL http://www.nd.edu/~oss/Data/data.html, 2005.

[10] G. Schwarz, "*Estimating the dimension of a model*". *Annals of Statistics* **6(2)**, pp. 461–464, Mar 1978.

[11] S. Staniford, V. Paxson, and N. Weaver, "*How to own the Internet in your spare time*". In 11th *USENIX Security Symp.*, pp. 149–167, Aug 2002.

[12] H. Wang and C. Wang, "*Open source software adoption: a status report*". *IEEE Softw.* **18(2)**, pp. 90–95, Mar/Apr 2001.

[13] G. K. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.