Sen, Ravi

Working Paper

Open Source Software Development Projects: Determinants of Project Popularity

EERI Research Paper Series, No. 2/2006

Provided in Cooperation with:
Economics and Econometrics Research Institute (EERI), Brussels

Suggested Citation: Sen, Ravi (2006) : Open Source Software Development Projects: Determinants of Project Popularity, EERI Research Paper Series, No. 2/2006, Economics and Econometrics Research Institute (EERI), Brussels

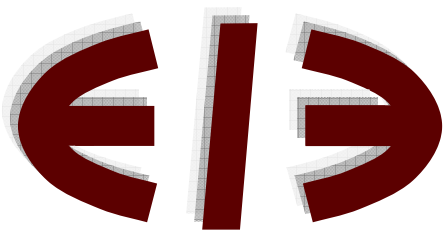
This Version is available at:
https://hdl.handle.net/10419/142507

EERI
Economics and Econometrics Research Institute

Open Source Software Development Projects: Determinants of Project Popularity

Ravi Sen

EERI Research Paper Series No 2/2006



EERI
Economics and Econometrics Research Institute
Avenue de Beaulieu
1160 Brussels
Belgium

Tel: +322 299 3523
Fax: +322 299 3523
www.eeri.eu

1. ABSTRACT

This paper is an initial exploration of the determinants of open source project success as measured by project popularity. We simultaneously model the impact of project-specific characteristics on project popularity, and the impact of intended users and choice of operating system on the choice of end-user license. These models are jointly estimated using Full Information Maximum Likelihood Method. The results show that the software-user license, age of the project, project status, certain types of potential users, and compatibility with certain operating systems have a statistically significant impact on project popularity. An interesting finding is that GPL, the most widely used software license has an adverse impact on the popularity of an open source project.

Keywords: Open source project, OSS, FLOSS, OSS popularity, OSS success

Open Source Software Development Projects: Determinants of Project Popularity

1.0 INTRODUCTION

The open source software movement has come a long way as is evident in the widespread adoption of open source software such as Linux and Apache. In a November 2003 CIO survey of 375 information executives, 54% said that within five years, open source would be their dominant platform (Koch 2003). A major segment of IT users, i.e. governments (e.g. Munich, Brazil, China, and Japan), are also adopting open source applications and encouraging open source projects (Fest 2001). These statistics suggest that open source software is here to stay. However, the fact remains that out of thousands of open source projects, only a few achieve the popularity of Linux or Apache. Most open source projects are small in size and are far less known. Nevertheless, it is important for all project development teams to understand the determinants of project popularity for the following three reasons-

a. **Attract Developers**- It has been suggested in the current literature that many developers participate in an open source project because they want to send out a signal for their advanced skills to their peers and/or potential employers (Lerner and Tirole 2002). For example, Hann *et al.* (2002) found that higher status within Apache Project was associated with significantly higher wages. If this is indeed the case, then potential developers are likely to favor associating with relatively more popular open source projects since these projects are more likely to provide the exposure that potential developers want in order to signal their programming skills to their peers and potential employers.

b. **Attract Non-Developer Users**- In the absence of any other information such as existing user base, potential users will need to install and tryout an open source application before they discover its quality. However, project popularity reassures them about the quality of software without having to install and use it first. Furthermore, popular software is more likely to have a large community of users to fall back on in case of technical problems.

c. **Attract Open Source Project Sponsors**- Since the emergence of the Internet, the cost of coordinating open source software projects has gone down. However, there are several factors that can stretch a project team's budgets. For example, a project may get into situations where relatively expensive testing equipment is required, or the project might require extensive and expensive patent research in order to avoid legal hassles at a later stage. Due to reasons such as these, many open source projects cannot progress beyond a certain point unless they receive additional financial support to handle these expenses. This additional financial support comes from individuals and organizations, and popular projects are likely to attract more donations.

In light of our discussion so far, a critical question that arises concerns the measure of success for an open source software. We now discuss this issue.

1.0 Measuring Open Source Project/Software Success

A popular measure of any software project's success is the installation base for that software. In addition to this measure, Crowston *et al.* (2003) identified three other measures of success for open source projects, namely project activity level, development team/community size (i.e. number of active contributors to the project), and time taken to

fix software bugs. They found a high correlation between project activity levels, community levels, and number of downloads indicating that these are measuring a common factor. To the best of our knowledge, there is no published research that investigates the adequacy and appropriateness of these measures. In order to maintain the focus of this paper, which is to investigate the determinants of project success, we choose the size of installation base as the measure of open source project success. The intuitive appeal of this measure and the availability of data for this measure also guided our choice.

The size of the installation base is easy to estimate in case of a commercial software since its market share is a good indicator of this measure. However, the market share information for most open source software is not readily available. In the absence of a reliable measure for an open source software's installation base, Freshmeat.net, which hosts information about more than 35,000 open source projects, uses a measure called project popularity. *Project Popularity* (denoted by *POPULAR* in this paper) of a project is a function of the number of times information is accessed about this project at Freshmeat's website, the number of hits (i.e. visits) received by the project's own website, and the number of subscribers of this project. It is calculated as follows:

*Project Popularity = ((record hits + URL hits) * (subscriptions + 1))^(1/2)*

As one can see, the formula places a high significance on the number of subscribers, who are more likely to be actual users of the software. A relatively low importance is given to record and URL hits because the hit could be a result of someone clicking on the project *link* by mistake, or the hit could be generated by people curious to

known about the project but may or may not actually use the software developed under this project.

It is one thing for the project development team to know that potential developers, software users, and/or project sponsors might be using this freely available measure to identify relatively more successful open source projects, however, what is at least as important from the project teams' perspective is to understand the influence of *project-specific* characteristics on project popularity. This understanding will help them to improve the popularity of their project, which is important for several reasons discussed earlier.

So far the research on the success of open software projects is very limited. Although some researchers have focused on defining the success measures of open source software (Crowston *et a.* 2003), to the best of our knowledge, we are not ware of any published research that investigates the determinants of the success of an open source project. Investigation of such determinants is further complicated by the potential endogeneity of some these factors in any model for project success.

In this paper, we use project-specific data from *Freshmeat.net* to empirically investigate the impact of *project-specific characteristics* (i.e. potential users, operating system, end-user license etc.) on *project popularity*, accounting for any potential endogeneity of some of these factors. Among others things, we find that the choice of GPL as the end-user license has an adverse impact on project popularity. This is a surprising result given the fact that more than 70% of open source software are released under GPL. Note that, when we do not account for the endogeneity of end-user license

choice in the model for software popularity, the results show that choice of GPL has a positive impact on project popularity.

The rest of the paper is organized as follows. Section 2 develops the model, Section 3 presents the model estimation and results, followed by Sect 4 that concludes the paper with a summary and discussion of the results.

2.0 MODEL FORMULATION

2.1 Indicator Variables

We wish to examine the affect of following project-specific characteristics on the project popularity.

Project Age: Intuition would suggest that age of the project should have a positive impact on the popularity of the project since the projects that are older had more time to attract interest from potential users, resulting in more hits to their project website and possibly a large number of subscribers. This construct is represented by AGE in the model and is measured in number days elapsed between the time this project registered at Freshmeat.net and 12/31/2004, the date when data used in this paper was accessed.

Software User License: Currently there are more than a few licenses under which open source software is distributed. Choice of end-user license impacts the number of potential subscribers, and therefore the popularity scores of a project. For example some subscribers might prefer licenses that given them a lot of freedom with the software (e.g. developers who want to integrate codes from different software), while others might not pay too much attention to the end-user license as long as it allows them to use the software for free (e.g. general desktop users). Learner and Tirole (2005) categorize open source licenses as highly restrictive (e.g. GPL) and relatively less restrictive (e.g. LGPL,

BSD). We use the same classification, and divide all the licenses (variable *LIC*) into two categories, namely GPL (i.e. *LIC* = 1) and OTHERS (i.e. *LIC* = 0).

Status of the Project: This construct, represented by *STATUS*, measures the progress of open source projects. A project can have one or more of the following status- Pre-Alpha (value =1), Alpha (value =2), Pre-Beta (value =3), Beta (value =4), Stable (value =5), where Pre-Alpha is the status of any project in its starting phases and Stable is the status when a project is ready for release. When a project has more than one status, it implies there are different versions of the same software in different stages of development. In such cases, the model computes an average score for the project status. We are interested in examining whether the status of an open source project has any influence its popularity of a project. One could argue that projects in the later stages (e.g. Stable) would have higher popularity score since more they are ready for use, while a project in its early stages of development will have a lower popularity since it will attract less interest and therefore fewer potential subscribers.

Target Users: Any software's intended users determine the size of the potential installation base for this software. Therefore, it would be interesting to examine the influence of the type of target user on an OSS project's popularity. This paper classifies the potential users of any open source software into six categories- Advanced Users (i.e. *A_USER*), Developers (i.e. *DEVLP*), Desktop Users (i.e. *D_USER*), Quality Engineers (i.e. *QENG*), System Administrators (i.e. *S_ADMN*), Others (i.e. *O_USER*). All these categories are represented as distinct dichotomous variables in the model because any software can be targeted at more than one type of user.

Operating System: Any software application has to be compatible with one or more operating systems. Therefore, the installation base of the target operating systems will definitely influence the number of potential subscribers and hence the popularity of an open source software. In this paper, operating systems are classified as- Windows (i.e. *WINDOW*), Macintosh (i.e. *MAC*), various flavors of Unix/Linux/POSIX (i.e. *UNIX*) and others (i.e. *O_OS*). Since software can be compatible with one or more of these operating systems, each of these options is represented as distinct dichotomous variables in the model.

2.2. Statistical Model

The following linear regression models the popularity of an open source project-

$$\ln(POPULAR_i) = \begin{bmatrix} \beta_{p0} + \beta_{p1}(AGE_i) + \beta_{p2}(LIC_i) + \beta_{p3}(STATUS_i) + \\ \beta_{p4}(A\_USER_i) + \beta_{p5}(D\_USER_i) + \beta_{p6}(QENG_i) \\ + \beta_{p7}(S\_ADMIN_i) + \beta_{p8}(O\_USER_i) \\ + \beta_{p9}(WIN_i) + \beta_{p10}(MAC_i) + \beta_{p11}(UNIX_i) + \beta_{p12}(O\_OS_i) + \varepsilon_i \end{bmatrix} \quad \textbf{(1)}$$

where ε_i, the random error term representing the unmeasured factors that impact the popularity of an open source project, is distributed as $N(0, \sigma_\varepsilon^2)$.

The model for project popularity includes some factors that could influence this outcome. However, many other factors that could potentially influence project popularity are not included in the model. The influence of these unmeasured factors is captured by the error term ε_i. Some of these factors could possibly influence the choice of the end user license (represented by the indicator variable *LIC*). For example, a survey done by The Boston Consulting Group found that the project leader's role influences a project's

success.[1] Project leader is also instrumental in choosing the end-user license for the software under development. Therefore, the choice of license is influenced not only by factors such as the target audience and the operating system on which the software runs (Lerner and Tirole 2005), but also potentially by the leader's preference/bias for one or the other end user license. Thus, the explanatory variable, *LIC*, might be potentially endogenous. In order to deal with this issue, we jointly model the choice of license by the project team with the popularity of open source project. We now develop the model for the choice of software user license.

Let the latent utility function of the project team/project leader of project i for the choice of end-user license be denoted by U_{iL}^{*}. We model U_{iL}^{*} as follows:

$$U_{iL}^{*} = \begin{bmatrix} \beta_{L0} + \beta_{L1}(A\_USER_i) + \beta_{L5}(D\_USER_i) \\ + \beta_{L6}(QENG_i) + \beta_{L7}(DEVLP_i) + \beta_{L8}(S\_ADMIN_i) \\ + \beta_{L9}(O\_USER_i) + \beta_{L10}(WIN_i) + \beta_{L11}(MAC_i) \\ + \beta_{L12}(UNIX_i) + \beta_{L13}(O\_OS_i) + z_i \end{bmatrix} = \mathrm{B}_L^{'} X_{iL} + z_i \qquad \textbf{(2)}$$

where X_l denotes the vector of covariates in U_{iL}^{*}, B_L denotes the associated vector of coefficients, and z_i denotes the random error term representing all the unmeasured factors that influence the utility from the choice of a software license. We assume that z_i has a standard normal distribution. The remaining covariates have their meanings as discussed earlier. We assume that a project opts for a GPL license if $U_{iL}^{*} > 0$ and OTHER license if $U_{iL}^{*} \le 0$. The probability that a project opts for a GPL license is given by:

$$P(LIC_i) = P(U_{iL}^{*} > 0) = P(\mathrm{B}_L^{'} X_{iL} + z_i > 0) = \Phi(B_L^{'} X_{iL}) \qquad \textbf{(3)}$$

[1] Based on a survey by The Boston Consulting Group and OSDN (http://www.osdn.com/bcg/).

Note that different sets of explanatory variables are included in different models. We only include those variables in a model that should impact the relevant outcome being modeled and that are available in the data. In other words, any variable that does not impact project popularity directly has been excluded from the model. Therefore, we do not include *DEVLP* in the model because our preliminary analysis suggests that *DEVLP* does not have a significant impact on *POPULAR*.

As discussed earlier, the explanatory variable *LIC* in the project popularity model is potentially endogenous. Ignoring this potential endogeneity and estimating the project popularity model independently would lead to biased estimates. One solution to deal with this issue is to use incomplete information methods (e.g. 2SLS). We, however, use full information maximum likelihood to estimate the model. This method estimates all the model parameters simultaneously and is more efficient. Thus, we model the choice of software license (called *LIC* Model in the rest of the paper) jointly with the model for project popularity (henceforth called *POPULAR* Model). A suitable correlation structure between the errors in the models makes the sources of correlations between the outcomes (*LIC* and *POPULAR*) explicit, i.e. part of the model, and hence the parameters estimated in the joint model estimation should be unbiased. To control for the correlation between the unmeasured factors, we assume that (ε_i, z_i) has the following bivariate normal distribution:

$$\begin{pmatrix} z_i \\ \varepsilon_i \end{pmatrix} \sim N\left(0, \begin{pmatrix} 1 & \\ \rho\sigma_z & \rho\sigma_z^2 \end{pmatrix}\right) \qquad \textbf{(4)}$$

In the joint model, the likelihood contribution of each project (joint likelihood of both the outcomes modeled for each project) can be written as the product of the

likelihood of observing the observed LIC_i, and then the conditional likelihood (conditional on LIC_i) of observing the observed project popularity (i.e. value of $POPULAR_i$). The first part contains the likelihood of obtaining the outcome of the probit model (i.e. L_{1i}), and the second part contains the likelihood of *POPULAR* value observed, conditional on the outcome of the probit model (i.e. L_{2i}). Therefore, the likelihood function for a project (i.e. L_i) becomes:

$$L_i = L_{1i} \times L_{2i} \qquad \textbf{(5a)}$$

Where

$$L_{1i} = \begin{cases} \Phi\left(\beta_L^{'} X_{iL}\right).....GPL \\ 1 - \Phi\left(\beta_L^{'} X_{iL}\right).....Non-GPL \end{cases} \qquad \textbf{(5b)}$$

and

$$L_{2i} = f(\varepsilon \mid z) \frac{1}{\sqrt{2\pi\sigma_{\varepsilon|z}^2}} \exp\left\{ -\frac{1}{2} \frac{(z - \mu_{\varepsilon|z})^2}{\sigma_{\varepsilon|z}^2} \right\} \qquad \textbf{(5c)}$$

$\mu_{z|\varepsilon}$ is the conditional mean of ε given z, and $\sigma_{\varepsilon/z}^2$ is the conditional variance of ε given z. The next section describes the data used in the research, model estimation, and results.

3.0 MODEL ESTIMATION AND RESULTS

We first estimate the models for project popularity and choice of software license independently. We then estimate the joint model for both outcomes and compare the results of the two estimations. We use Full Information Maximum Likelihood (FIML) to estimate the model.

3.1 Data

The data consist of all open source projects listed at Freshmeat.net. Freshmeat is a free service that maintains a large database of software applications, which are preferably released under an open source license. For each project, the database provides a description of the software, links to download it and to obtain more information, and a history of the project's releases. In addition it also provides news on new releases, and offers a variety of original content on technical, political, and social aspects of software and programming. The site is funded by VA Software, a leading provider of software, information, and community support to IT (information technology) managers and software development professionals.

The Freshmeat database contained (as of December 2004 when the data used in this study was obtained) approximately 35,000 software projects. The data contains information about all projects registered since January 1998 till 17th December 2004. For the purpose of this study we considered only those projects for which complete information was available. The number of such projects was 12,923.

The data includes information about the date a project was registered with Freshmeat, the type of end-user software license, the operating system under which the software would run, and potential users of the software. The data available at Freshmeat is reported by project teams and therefore, one can question their accuracy. Learner and Tirole (2005) provide a possible reason for not suspecting the accuracy of this data. They explain "*.....the project leaders are trying to recruit new developers, attract new users, and solicit donations for their project. Undertaking a "bait-and-switch" strategy to do so, e.g. making the project appear something other than what it really is, is unlikely to be*

a positive signal for prospective developers, users and/or sponsors." *Table 1* presents the summary statistics of the dataset used in this study.

TABLE 1: Data Summary

Variable		Mean	Std Dev	Min	Max
POPULAR	(Project Popularity)	413.24	989.81	4.36	39298.17
AGE	(Age of the Project)	933.36	600.04	17.30	2545.13
LICENSE	(License for Software Use)	0.72	0.45	0	1
STATUS	(Status of the Project)	4.39	0.91	0	1
AUSER	(Advanced User)	0.04	0.19	0	1
DEVLP	(Developer)	0.47	0.50	0	1
DUSER	(Desktop User)	0.52	0.50	0	1
QENG	(Quality Engineer)	0.03	0.18	0	1
SADMIN	(System Administrator)	0.30	0.46	0	1
OUSER	(Other User)	0.06	0.24	0	1
WINDOW	(Windows)	0.46	0.50	0	1
MAC	(Macintosh)	0.39	0.49	0	1
UNIX	(Flavors of Unix, Linux)	0.98	0.16	0	1
O_OS	(Other Operating System)	0.36	0.48	0	1

As we can see from the summary statistics (Table 1), about 50% of projects are in later stages of development, most of the projects use GPL for the software that they develop (72%), and developers and desktop users are most popular intended audience for the software being developed. 90% of the software listed on Freshmeat's runs on various flavors of Unix and Linux, 46% of the software run on Windows and about 30% run Macintosh. *Figure 1* presents the distribution of popularity scores of open source projects.

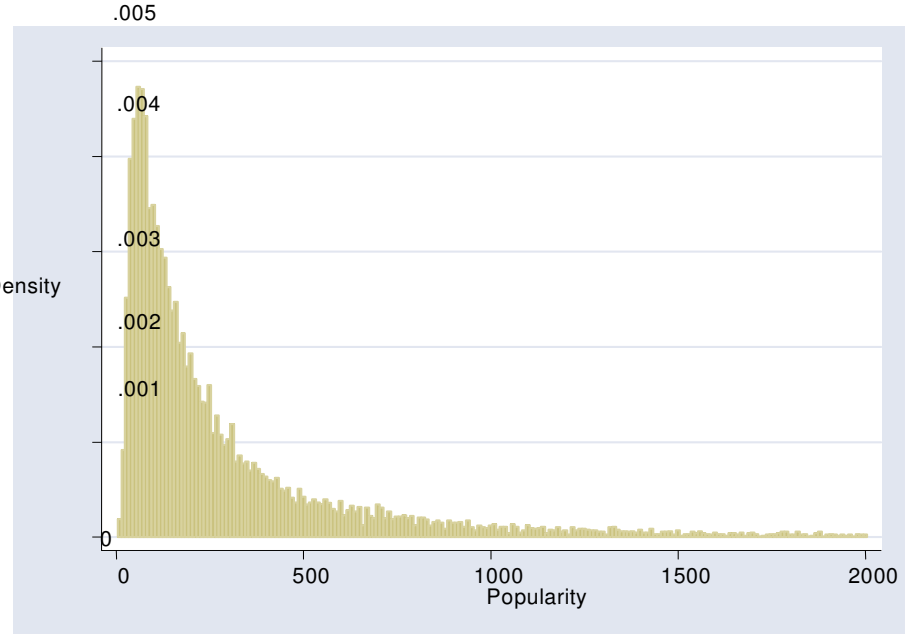


Figure 1: Shows the frequency distribution of project popularity scores for scores less than 2000.
[Note: The tail of the distribution extends all the way to the maximum popularity score. This part of the graph is not shown for clarity].

3.2 Model Estimation

Table 2 shows the joint and independent model estimates for the *POPULAR* and *LIC* models.

TABLE 2: Model Estimates

	Joint Estimation	**Independent Estimation**		**Joint Estimation**	**Independent Estimation**
POPULARITY MODEL (Linear Regression)			**LIC Model** (Probit)		
CONSTANT	2.1985 * (0.1138)	1.5524 * (0.1083)	**CONSTANT**	0.0673 (0.0801)	0.0631 (0.0800)
LN-AGE	0.3941 * (0.0118)	0.3992 * (0.0120)	**AUSER**	-0.0711 (0.0625)	-0.0485 (0.0626)
LICENSE	-1.0246 * (0.0656)	0.1986 * (0.0210)	**DEVLP**	-0.1071 * (0.0249)	-0.1244 * (0.0290)
STATUS	0.1234 * (0.0108)	0.1254 * (0.0108)	**DUSER**	0.2671 * (0.0265)	0.2601 * (0.0278)
AUSER	0.1029 (0.0652)	0.1309 * (0.0597)	**QENG**	-0.3010 * (0.0665)	-0.2959 * (0.0673)
DUSER	0.2748 * (0.0220)	0.1476 * (0.0190)	**SADMIN**	-0.1243 * (0.0271)	-0.1409 * (0.0279)
QENG	0.0396 (0.0589)	0.1943 * (0.0534)	**OUSER**	0.0507 (0.0506)	0.0429 (0.0507)
SADMIN	0.3413 * (0.0228)	0.3781 * (0.0203)	**WINDOW**	-0.3850 * (0.0379)	-0.3985 * (0.0385)

OUSER	-0.073 (0.0434)	-0.0923 * (0.0397)	**MAC**	-0.2774 * (0.0444)	-0.2796 * (0.0457)
WINDOW	0.0549 (0.0335)	0.2226 * (0.0298)	**UNIX**	0.6879 * (0.0773)	0.7119 * (0.0763)
MAC	0.0764 (0.0392)	0.1980 * (0.0354)	**O_OS**	0.2899 (0.0444)	0.3073 * (0.0458)
UNIX	0.4376 * (0.0728)	0.1289 (0.0699)	σ_z	1	1
O_OS	-0.1296 * (0.0395)	-0.2613 * (0.0352)	ρ	0.6337 * (0.0238)	0
σ_ε	1.1532 * (0.0148)	1.0207 * (0.0065)	Log-Likelihood	-25926.64	25956.38

NOTE: Asymptotic standard errors in parentheses;
Significance: *'= 5%

From the results it is clear that in the joint estimation, the parameter ρ, which is the correlation between the unmeasured heterogeneity terms in the models for *POPULAR* and *LIC* is significant (ρ = 0.6337*). This confirms the endogeneity of *LIC* in the model for *POPULAR*. The positive value of this correlation implies that the unmeasured factors that increase the probability of the project team choosing GPL as end-user license also increase the popularity of the project, i.e., impact both the utility from the choice of GPL as end-user license, and the project popularity in the same direction.

While discussing the estimates we'll now consider only the estimates from the joint model estimation. The joint estimation of *POPULAR* model show that the coefficients for *AUSER* (advanced users), *QENG* (quality engineering), *OUSER* (other users), *WINDOW* (software that run on Windows), and *MAC* (software that run on Macintosh) are not significant. Similarly, the joint estimation for *LIC* model show that coefficient for *AUSER* (advanced users), *OUSER* (other users), *O_OS* (software that run on operating systems other than Unix/Linux, Windows, and Macintosh) are also not significant. Therefore, we'll ignore these variables in subsequent discussions.

3.3 POPULAR Model

We find that the choice of end-user license (*LIC*), the age of the project (*AGE*), the status of the project (*STATUS*), operating system (i.e. *UNIX*), and certain types of intended users (i.e. *DUSER* and *SADMIN*) have a significant impact on the popularity of the project.

As expected, we find that the choice of end-user license (*LIC*) is endogenous. The joint model estimation shows that if a project chooses GPL as the end-user license for its software, the popularity of the project decreases by approximately 64% $[= (1 - e^{-1.0246}) \times 100]$. This is an interesting result if we take into account the fact that GPL is the most widely used end-user license among the open source project community. Note that the choice of GPL has a positive impact on project popularity if the *POPULAR* Model is estimated independently (i.e. the endogeneity of *LIC* is not taken into account). According to this result the popularity of open source project increases by approximately $22\% [= (e^{0.1986} - 1) \times 100]$, when the project leader(s)/owner(s) choose GPL as the end-user license for the open source software.

This adverse impact of GPL on the popularity of an open source project could be attributed to the restrictive nature of GPL. The GPL permits free use, modification, and redistribution of software and its source code by anyone, but imposes certain key restrictions. For instance, if a licensee includes any amount of GPL code in another program, that entire program becomes subject to the terms of the GPL. Certain types of potential users might prefer a less restrictive end-user license (e.g. BSD), since they might want to retain the right to set the terms for the reuse of the software code in a way

that best serves their objectives. Therefore, the popularity of open source software with GPL as end-user license will suffer among these users.

Moving on to other indicator variables, an increase of 10% in the age of the project (i.e. *AGE*) increases the popularity of the project (i.e. *POPULAR*) by approximately $4\%[=(1.1^{0.3941}-1)\times 100]$. As a project (i.e. *STATUS*) moves up one status, its popularity increases by approximately $13\%\ [=(e^{0.1234}-1)\times 100]$. Projects that develop software targeted at desktop users (*DUSER*) are $32\%\ [=(e^{0.2748}-1)\times 100]$ more popular than projects targeted at non-desktop users. Similarly, projects targeted at systems administrators (*SADMIN*) are approximately $41\%\ [=(e^{0.3413}-1)\times 100]$ more popular than those targeted at others.

Finally, software that run on Unix-like platforms (*UNIX)* are about 55% more popular than projects that do not run on these platforms. This result is along expected lines since most Unix and later various flavors of Linux are the flagship operating systems for the open source community of developers. Interestingly, the independent estimation of *POPULAR* model shows *UNIX* to have an insignificant impact on project popularity, while *WINDOW* and *MAC* to have a significant impact on project popularity.

3.4 *LIC* Model

All potential users, except *AUSER* (advanced users) and *OUSER* (other users), have a significant impact on the choice of software license. Similarly, *WINDOW, MAC, UNIX* have a significant impact on the choice of end-user license. This result is in line with an earlier finding by Lerner and Tirole (2005), who found that the operating system on which the software will run has an impact on the choice of end-user license.

The probability of choosing GPL as the end-user license increases by approximately 17% if the intended users of the open source software are desktop users. On the other hand, the probability of choosing GPL decreases by approximately 9% if the intended users are developers, decreases by approximately 29% if the intended users are quality engineers, decreases by approximately 10% if the intended audience is systems administrators. Interestingly, the last result is different from the one obtained by Learner and Tirole (2005. They found that software applications geared towards system administrators are more likely to have a restrictive license such as GPL.

These results can be explained by the restrictive nature of GPL as explained earlier. Developers, quality engineers and systems administrators are highly skilled software professionals who might want to retain the right to set the terms for the reuse of the software code in a way that best serves their objectives. A less restrictive license (e.g. BSD) would allow them more flexibility in tailoring the open source software to their own unique needs. GPL, on the other hand, restricts them in terms of whether they can integrate the software with any other software that is used under any other license.

The probability of choosing of GPL increases by approximately 32% if the operating system on which the open source software will run on some flavor of Unix and Linux. However, the probability of choosing GPL decreases by approximately 40% if the software will run on Windows and by approximately 26% if the software will run on Macintosh. These results are similar to ones obtained by Learner and Tirole (2005), who found that "*restrictive licenses (such as GPL) are less common for projects operating in commercial environment or that run on proprietary operating systems*." This is not surprising since advocates of BSD end-user license often argue that GPL and related

licenses discourages potential commercial users since GPL is restrictive in terms of the complementary software (specially those released under commercial end-user license) that could be used with the open source license released under GPL license.

4. 0 SUMMARY AND DISCUSSION

This paper empirically investigates the influence of project-specific characteristics on the popularity of open source projects. We use data on open source projects registered at Freshmenat.net to jointly estimate a log-linear regression model for project popularity and a probit model for the choice of end-user license. The joint model estimation shows that projects, which develop software that run on Unix-like operating systems, are more popular. On the other hand, the choice of any other operating system (such as Windows and Macintosh) has no significant impact on project popularity. We also find that projects targeted at desktop users and systems administrators are relatively more popular. Finally the age of a project and its development status also have a significant impact on the popularity of the project.

The most interesting result, however, is the significant but adverse impact of GPL end-user license on the popularity of an open source project. GPL is the most popular license among open source projects. The GPL permits free use, modification, and redistribution of software and its source code by anyone, but imposes three key restrictions on every licensee: (a) the licensee distributes code licensed under the GPL, it must guarantee availability of the source code for the entire work for unlimited replication to anyone who wants it; (b) when the licensee distributes GPL code, it may not charge a licensing fee or royalty for the software, but may charge only for the cost of

distribution; and (c) if the licensee includes any amount of GPL code in another program, that entire program becomes subject to the terms of the GPL. Many intended users of open source software (e.g. system administrators, and developers) are most likely to use several software applications in their operating environment. They would prefer a software license such as BSD or Apache-style licenses, which allows them to integrate open source software with software released under other licenses, and still be able to use and share this integrated software under a license of their choice without worrying about the legal ramifications of this action.

However, we believe that despite the adverse impact of GPL on project popularity, its use will not decline in the near future. As one of the results shows, the GPL will be the most likely choice for open source software targeted at desktop users. These users generally use the software on an "as is" basis. They usually do not make modification to the software code or integrate it with other software, and therefore, they are not much concerned about end-user license. On the other hand, skilled software professionals (i.e. developers, systems administrators etc.) prefer less restrictive licenses and therefore, are very much concerned about the end-user license under which an open source software is released. Open source project leaders are aware of this concern and their response is evident in one of the results, which shows that the likelihood of choosing GPL decreases if the software is targeted at these professional users. However, some open source projects, even though targeted at skilled users, would continue to prefer the GPL because - (a) the software benefits from the network affect generated by the large number of open source released under GPL license (e.g. a choice of GPL license increases the likelihood that a software will be able to find another complementary

software also released under GPL); (b) GPL helps their "fight" against commercialization of software since it ensures that the open source software (released under GPL) is not combined with a commercially licensed products and then released under a commercial end-user license; (c) choice of GPL as end-user license attracts potential developers who strongly believe in the GPL philosophy; and/or (d) if they do not use GPL, other open source developers may start a competing project under GPL.

This paper leaves a number of issues unaddressed, which could be investigated in future research. For example, the indicator variables in the model for project popularity do not include characteristics specific to project developers. The impact of these characteristics on project popularity could be highly revealing, especially if their simultaneous impact on the choice of end-user license is also factored into the analysis. Another interesting area that needs further investigation is the identification of factors that could influence other measures of open source project success that are currently in use (e.g. see Crowston *et al.*) and their actual impact on open source project success.

REFERENCE

1. Crowston, K., Annabi, H., and Howison, J. 2003. Defining Open Source Software Project Success. In *Proceedings of the 24th International Conference on Information Systems (ICIS 2003),* Seattle, WA.
2. Fest, P. 2001. Governments push open-source software. *News.com*, August 29.
3. Gaudeul, A. 2003. The $(L^A)T_E X$ Project: A Case Study of Open-Source Software. *TUGBoat*, (24), pp. 1001-1015.
4. Hann, Il-Horn, Robert, J. and Slaughter, Sandra A. 2004. Why developers participate in open source software projects: an empirical investigation. *Proceedings 25th International Conference on Information Systems*, 13-15th December, pp. 821-830.
5. Johnson, J.P. 2002. Open source software: Private provision of a public good. *Journal of Economics and Management Strategy*, 11(4), pp. 637-662.
6. Koch, C. 2003. Your Open Source Plan. *CIO*, March 15.
7. Lerner, J. and Tirole, J. 2002. Some Simple Economics of Open Source. *Journal of Industrial Economics*, 50, pp. 197-234.
8. Lerner, J. and Tirole, J. 2005. The Scope of Open Source Licensing. *Journal of Law, Economics, and Organization*, April, 21(1), pp. 20-56.
9. Moon, J.Y., and Sproull, L. 2000. Essence of Distributed Work: The Case of the Linux Kernel" *Firstmonday*, 5(11).
10. Mustonen, M. 2003. Copyleft- The Economics of Linux and Other Open Source Software. *Information and Economics Policy*, 15(1), pp. 99-121.
11. Raymond, Eric S. 1999. *The Cathedral and Bazaar*, O'Reilly: Sebastopol, California.