

Consumption and Production of Digital Public Goods

Modeling the Impact of Different Success Metrics in Open Source Software Development

Nicholas P. RADTKE^{1,3} and Marco A. JANSSEN^{2,3}

Abstract—With the Internet has come the phenomenon of people volunteering to work on digital public goods such as open source software and online encyclopedia articles. Presumably, the success of individual public goods has an effect on attracting volunteers. However, the definition of success is ill-defined. This paper explores the impact of different success metrics on a simple public goods model. The findings show that the different success metrics considered do have an impact on the behavior of the model, with the largest differences being between consumer-oriented and producer-oriented metrics. This indicates that many proposed success metrics may be mapped into one of these two categories and within a category, all success metrics measure the same phenomenon. We argue that the characteristics of producer-oriented metrics more closely match real world phenomena, indicating that public goods are driven by producer, and not consumer, interests.

Index Terms—Digital public goods, success metrics, FLOSS, open source software, Wikipedia.

1. INTRODUCTION

IN recent years an interesting phenomenon can be observed in the digital media. People are volunteering their time to contribute, for example, to the creation of software [1] or an online encyclopedia [2] at their own costs for the benefit of the wider population. Such products can be called public goods. Traditional non-cooperative game theory argues that people will not invest in public goods since it benefits to free ride on the investments of others.

Research in psychology, economics, and political science shows that people invest in public goods, as observed in case studies and replicated in controlled experiments with human subjects [3]. A variety of factors are put forward as possible explanations for these contributions, such as other-regarding preferences, communication, etc.

Another finding is that the level of cooperation reduces as group size increases [4]. Therefore, it is remarkable to see high levels of contributions to digital public goods like open source software and Wikipedia, where the number of people involved is huge. Looking into more detail of the statistics, it

can be seen that the distribution of the public goods products which are successful is skewed, as is as the distribution of producers working on public goods. For example, only 14% of the projects at SourceForge, the largest site hosting open source projects, have been updated during the last year.⁴

In an earlier paper [5] we presented an empirically grounded model that captures several main patterns of the SourceForge repository of open source data. One of the assumptions is that the success of a project affects the attractiveness of a project. However, a key problem in the literature of open source software is the ambiguity of the definition of success for a project. Can it be measured by the number of downloads, the frequency of releases, the number of bug fixes, or any number of other indicators?

This paper presents a more theoretical and simpler model than [5] of the evolution of populations of digital public goods; the model is used to test the consequences of different definitions of success.

First, basic empirical findings from studies on open source software and Wikipedia will be presented in Section 2. In Section 3 the model will be presented, and the analysis of the model is contained in Section 4.

2. EMPIRICAL PATTERNS

With Web 2.0 people can contribute and consume goods which are freely available to others. What makes some of these products successful and others not? How is success defined?

2.1. SourceForge

SourceForge is a site that contains a set of online tools facilitating the development of open source software. It was established in November, 1999 and as of November, 2008 hosts 138,674 projects. Using data from this site, the distribution of developers working on a project has been shown to be highly skewed, with 67% of projects having only one developer and 90% of projects having fewer than 4 developers [6]. It has been found that 10% of the developers write 72.3% of the code [7] and the 100 most active developers are involved in 1,886 different projects [8]. Also, the number of people reporting bugs is an order of magnitude greater than the number of developers fixing bugs, which is an order of magnitude greater than the number of core developers for a project [7].

¹School of Computing and Informatics, Ira A. Fulton School of Engineering, Arizona State University, P.O. Box 878809, Tempe, AZ 85287-8809, U.S.A.

²School of Human Evolution and Social Change, Arizona State University, P.O. Box 872402, Tempe, AZ 85287-2402, U.S.A.

³Center for the Study of Institutional Diversity, Arizona State University, P.O. Box 872402, Tempe, AZ 85287-2402, U.S.A.

⁴<http://www.SourceForge.net> year period defined as 11/28/07-11/27/08.

Various extensive surveys have been performed where open source developers are asked why they participate in open source software development (e.g., [9], [10]). The main reasons given are:

- Develop new skills
- Share knowledge with other developers
- Improve existing open source projects
- Engage in a new form of cooperation
- Enjoy the challenge
- Improve job opportunities

2.2. Wikipedia

The online encyclopedia Wikipedia started in January, 2001 and now⁵ consists of 11.8 million articles in 264 languages from 597 million edits by 14.6 million users. The distribution of contributions is skewed with 90% of the users contributing fewer than 10% of the edits [11]. The number of people per article and the number of edits per article follows a power law distribution [12].

[2] distinguishes factors that motivate people to participate, namely reputation and commitment to group identity, in Wikipedia. Registered users are assumed to be motivated more than anonymous users to contribute and to have higher quality contributions. [2] shows that anonymous users provide infrequent contributions, but the contributions are of high quality.

In summary, both digital public goods examples show that there is a skewed distribution of contributions.

2.3. Success Metrics

In order to understand why the contributions to SourceForge and Wikipedia are so skewed, it is necessary to understand why people contribute or use certain digital public goods. A possible explanation is that contributors prefer to participate in successful projects. However, there is no agreement how to measure the success of digital public goods. For example, success of open source software is not clearly defined. While there are no generally agreed upon standards, the following success indicators have been proposed:

- Completion of the project [13]
- Progression through maturity stages [14]
- Number of developers
- Level of activity (i.e., bug fixes, new feature implementations, mailing list)
- Time between releases
- Project outdegree [15]
- Active developer count change trends [15]

Furthermore, [16] asked eight developers how they defined success and failure of an open source project. Answers varied for success, but all agreed that a project with a lack of users was a failure. Thus having a sufficient user-base may be another metric for success.

⁵http://meta.wikimedia.org/wiki/List_of_Wikipedias accessed November 24, 2008.

For Wikipedia articles, success may relate to the quality of the articles. Quality of the articles is suggested to be related to the number of edits and unique editors to an article [17]. Usual manual evaluations of articles, factual accuracy [18], and credibility [19] are mentioned. Statistics of consumers and consumer experiences with articles would be helpful, but information on how many times articles have been read is not available.

In summary, there is no clear method to define success of digital public goods, especially for the purpose of modeling the development of these goods. Therefore we will use different definitions of success in our model that reflect the accumulation of activities and the number of users involved and explore whether different definitions of success have an impact on the patterns generated by the model.

3. MODEL DESCRIPTION

The model we present is a very simplistic model of consumers and producers of an ecology of public goods based on the observed processes of open source software development. Given are N_a agents and N_p projects. At each time step an agent may 1) contribute to the development of a project and/or 2) consume (a.k.a. use) a project. Each agent has a probability p_c to contribute to a project and a probability p_u to consume from a project. The probabilities p_c and p_u are drawn from an exponential distribution with parameter value 10. This represents the notion that most agents will have a small probability to be active during a time step. If a value higher than 1 is drawn, this result is ignored and a new value is drawn.

When an agent is active during a time step it will make a decision about which project to contribute to or use based on how close the characteristics of a project match with the preferences of the agent. To define how well agent preferences match characteristics of a project, the matching interests value M_i is calculated for each project, as shown in Equation (1):

$$M_i = 1 - (n_p - n_a)^2 \quad (1)$$

where n_p is the characteristics (a.k.a. needs) of the project and n_a the preference value for the agent.⁶ If both dimensions match, M_i is equal to 1. Initially values for n_p are assigned randomly from a uniform distribution. However, it is assumed that consuming agents may have certain needs (e.g., an interest in well-documented, easy-to-use projects) while producing agents will have a different set of needs (e.g., an interest in projects for the challenge and to gain experience) [9], [10]. Thus values for n_a are based on an agent's producer number p_c and consumer number p_u . A simple function for mapping from p_c and p_u to n_a is shown in Equation (2):

$$n_a = f(p_u, p_c) = \frac{p_c - p_u + 1}{2} \quad (2)$$

A visual depiction of the mapping is shown in Figure 1. Essentially, an agent's needs can be thought of as a continuum between 0.0 and 1.0. Lower values (< 0.5) represent a bias

⁶To keep consistent with our earlier publication [5] covering a more complex model, we refer to n_p and n_a as needs vectors. In this simplified model, these are one-dimensional vectors and can thus be treated as scalars.

