

December 2007

The Application of DEA to Measure the Efficiency of Open Source Security Tool Production

Richard Mathieu
James Madison University

Barry Wray
University of North Carolina at Wilmington

Follow this and additional works at: <http://aisel.aisnet.org/amcis2007>

Recommended Citation

Mathieu, Richard and Wray, Barry, "The Application of DEA to Measure the Efficiency of Open Source Security Tool Production" (2007). *AMCIS 2007 Proceedings*. 118.
<http://aisel.aisnet.org/amcis2007/118>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in AMCIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

The Application of DEA to Measure the Efficiency of Open Source Security Tool Production

Barry A. Wray
Cameron School of Business
University of North Carolina Wilmington
wrayb@uncwil.edu

Richard G. Mathieu (* corresponding author)
College of Business
James Madison University
mathierg@jmu.edu

Abstract

There are a wide variety of open source security tools available for deployment within the enterprise. Despite the success of many security-based open source software (OSS) projects, large numbers of these projects become inactive and are eventually abandoned. The purpose of this research is to develop an empirical study to determine the relative efficiency of security-based OSS projects. A security-based OSS projects utilize two key resources in their development, the number of software developers and the number of users submitting bugs found when using OSS code. This research develops a model to measure the relative efficiency of each project and determines the number of inefficient projects benchmarking each efficient project. The result of this research is a model that can be used by project developers to evaluate the relative efficiency of their projects. Our empirical study will be based on the analysis of publicly available data from a repository of OSS project at *sourceforge.net*.

Keywords: open source software, software production, DEA

The Application of DEA to Measure the Efficiency of Open Source Security Tool Production

Introduction

Open source software (OSS) projects permit users the freedom to use the software code for any purpose. The code can be studied, modified, and freely redistributed. Even though OSS is free, the profit potential of OSS projects is becoming very attractive to software development companies. Venture capitalists have pumped nearly \$400 million into 50 open-source companies in the last 18 months -- and more are on the way (Lacey, 2005). These products are satisfying business customer needs and giving birth to for-profit companies like SugarCRM, Greenplum, and Pentaho. These companies are building a new generation of business applications for managing Web content, customer relations, and enterprise resources that are cheaper and may be more dynamic than their commercial counterparts (Greenemeier, 2005). The potential financial gain for a developer lies in the support/maintenance and proprietary add-on features they can provide for their product.

The success of open source software (OSS) projects has been attributed to the quality, portability and scalability of the software product (Stemelos et al., 2002; Crowston et al., 2003; Kalina and Czyzycki, 2005) and to the commitment, expertise and speed of development of the software developers (Scacchi, 2002; Crowston et al., 2003). Despite the success of many OSS projects, large numbers of these projects become inactive and are eventually abandoned. Payne (2002) and Salkever (2001) examined the security of OSS projects. Crowston et al. (2003) developed a model of OSS success based upon measures of project output, process, and outcomes for project members.

A common way of examining the relative efficiency of software projects is through the use of data envelope analysis (DEA). DEA is a nonparametric technique which takes input and output data related to individual operational units and identifies an efficient frontier representing optimal performance as a ratio of output to input. DEA is well suited for the comparison and benchmarking of similar operational units. The result is a ranked list of operational units in terms of their relative efficiency and indicators of the variables with the largest influence on the operational unit rankings. In the software management literature, DEA operational units are typically defined as software projects. The DEA results have been used to determine the most productive scale size for a software development project (Banker and Kemerer, 1989) and to study the effects of project characteristics on software maintenance (Banker et al., 1991; Elam, 1991; Paradi et al., 1997).

Open Source Security Tools

There are a wide variety of open source security tools available for deployment within the enterprise. Currently there are quite a few mature and well-developed open-source security tools that are on par with commercial security tools (Mogull and Girard, 2006). These OSS security tools are typically publicly visible in all phases of their lifecycle and have many contributors. Mookhey (2004) defines nine categories of open source tools for security and control assessment (vulnerability assessment tools, network auditing tools, host-based auditing tools, password cracking tools, forensic tools, log analysis tools, software auditing tools, web application testing tools, and process auditing tools) that can be useful to information security auditors.

Three categories of open source security tools have reached the 'plateau of productivity' in Gartner's Hype Cycle. Specifically, open source intrusion detection software (Snort IDS), web security and encryption software (OpenSSL), and vulnerability assessment software (Nessus) has demonstrated and accepted benefits and are broadly accepted in the marketplace (Drakos et al., 2004). All three open source software tools are considered 'early mainstream' maturity with market penetration estimated between 5% to 50% of the marketplace. A recent Gartner Report (Mogull and Girard, 2006) notes that many open source security tools are initially created, but that only a few mature to the point where they have increased documentation, ease of use and functionality. All open source security tools seem to follow a consistent timeline in terms of maturity and public visibility.

Data Envelope Analysis

A common way of examining the relative efficiency of software projects is through the use of Data Envelopment Analysis (DEA). DEA is a nonparametric linear programming formulation technique that takes multiple inputs and multiple outputs related to individual operational units and identifies an efficient frontier representing optimal performance as a ratio of output to input. DEA is an extreme point method that compares decision making units or DMUs (the DMUs for this research are security-based OSS projects) with only the "best" DMUs. Each DMU must utilize the same inputs to produce the same outputs in order for the DEA model to evaluate the relative performance of a set of DMUs. The DEA model will produce a ranking of the DMUs according to how efficiently each DMU utilizes its inputs to produce its outputs.

As the earlier list of applications suggests, DEA can be a powerful tool when used wisely. A few of the characteristics that make it powerful are:

- DEA can handle multiple inputs and multiple outputs for each operational unit.
- No assumption of a functional form relating inputs to outputs is necessary
- Projects are directly compared against peer operational units or a combination of peer operational units.
- The model inputs and outputs can be in different operational units.

There are certain conditions that limit the use of DEA. This tool should not be used if any of the following conditions exist:

- Noisy data can cause significant problems.
- DEA is good at estimating "relative" efficiency of the operational units but it does not allow comparison to a theoretical maximum.
- Large problems can be computationally intensive.

In previous DEA studies of traditional software projects, operational unit input has been captured as a measure of labor hours or cost. One study also captured additional input measures of project expense (Elam, 1991). Measures of output have focused on project size through either function points or lines of code. Other measures of output include software quality and time to market. See Table 1.

Table 1: Review of Prior DEA Studies – Traditional Software Development

Study	Input: Labor	Input: Other Expenses	Output: Function Points	Output: Lines of Code	Output: Quality	Output: Time to Market
Banker and Kemerer, 1989	X			X		
Elam, 1991	X	X	X		X	
Banker, Datar and Kemerer, 1991	X		X	X		
Paradi, Reese and Rosen, 1997	X	X	X		X	X

From a project management perspective, there are significant differences between an open source software project and a traditional software project. First, since labor is typically donated by project contributors there are typically no monetary measures of labor cost or project expense. Rather the ‘labor’ that goes into an OSS project can be viewed as the number of persons that contribute to the project. Hahn and Zang (2005) adjust their input through an assessment of ‘project age’. A contribution of our research study is the inclusion of a count of the number of unique “software bug” contributors. These contributors are typically not on the development team but are users of the software making valuable contributors to the success of the project.

The output of an Open Source software project should certainly contain a measure of project size. Hahn and Zang (2005) measure the size of all files in the project. In this study we take a user oriented approach to project size by including a measure of project size per download. Hahn and Zang (2005) include a measure of ‘development status’. This study includes two measures of project quality: number of downloads and project rank (as determined by *sourceforge.net*).

Table 2: Review of DEA Studies – Open Source Software (OSS) Projects

Study	Input: Number of Developers	Input: Project Age	Input: Number of Bug Contributors	Output: Project Size	Output: Quality	Output: Development Status
Hahn and Zang, 2005	X	X		X		X
this research	X		X	X	X	

Objectives of Research

The purpose of this research is to develop and test a model of the relative efficiency of OSS security based projects. The focus on one type of OSS project is driven by the fact that DEA is well suited for the comparison and benchmarking of similar operational units. Focusing on one type of project reduced the variance in the technical characteristics of the software development projects and made for better comparisons between projects. Our empirical study will be based on the analysis of publicly available data from a repository of OSS projects at *sourceforge.net*. Currently *sourceforge.net* contains a repository of 142,869 OSS projects.

Research Methods

This research evaluates the relative efficiency of security-based OSS projects by evaluating multiple project inputs and multiple project outputs. Data were collected on 35 security-based OSS software projects on *Sourceforge.net* in August, 2006. The data were manually tabulated from the *Sourceforge.net* website and entered into an Excel spreadsheet. For brevity reasons data collected for only the 5 highest ranked projects are given in Table 3 (data for the remaining 30 projects is similar). The inputs considered for the 35 projects are the total number of developers for the project and the number of unique users that have submitted software bugs. Unlike traditional software projects, the number of bug submitters is an important input into the OSS production process. The outputs for each project are the *Sourceforge.net* rank for the project, the number of downloads from *Sourceforge.net*, and the number of Kilobytes per download.

Table 3: Security-Based OSS projects Collected from *Sourceforge.net*

	Bug {IN}	people {I}	rank {O}	downloads {O}	Kperdownload {O}
Open Computers and Software Inventory	66	7	4	39,425	35,510
Endian Firewall	24	10	18	9,033	106,310
KeePass Password Safe	44	8	27	82,098	688
Password Safe	64	29	34	34,925	1,354
Ophcrack	1	4	36	66,146	338,645

Data envelopment analysis (DEA) is a linear programming formulation for frontier analysis that defines a nonparametric relationship between multiple outputs and multiple inputs by building an efficiency frontier (Charnes et al., 1978). DEA is an extreme point method that compares decision making units or DMUs (the DMUs for this research are security-based OSS projects) with only the "best" DMUs. The efficient security-based OSS projects have an efficiency score of one whereas the inefficient projects have an efficiency score less than one but greater than zero. The EMS (efficiency measurement system) software version 1.3 (Scheel, 2000) is used in this research. This software can be found at: <http://www.wiso.uni-dortmund.de/lsg/or/scheel/ems/>.

The number of developers is a "standard" input for the DEA model while the number of unique bug submitters is a "non-discretionary" input (i.e., data which are not controlled by a project). The DEA model evaluates and compares these inputs and produces an efficiency score for each project based on project outputs. The method of producing an efficiency score is based on the convex envelopment technology structure and the efficiency measure chosen. The efficiency measure quantifies a "distance" to the efficient frontier.

The orientation for the DEA model used in this research utilizes an “input-oriented” measurement for efficiency (a measure of input reduction necessary for a project to become efficient holding the outputs constant). This measure is chosen because of the primal interpretation of the efficiency score with respect to the input quantities and the axiomatic properties of the efficiency measure.

Two distance measurements are used and the resulting project efficiencies are compared. The notation below uses τ to denote the technology used and (X^k, Y^k) to denote the input output data of the project under evaluation. The first measure is the Debreu-Farrell “radial” measure (Farrell, 1957). This measure indicates the necessary improvements when all relevant factors are improved by the same factor equiproportionally. The objective for this model is:

$$\min \{ \theta \mid (\theta X^k, Y^k) \in \tau \}$$

The second model chosen is a “minAverage” measure that quantifies the minimal average of relative improvements necessary for a project to become weakly efficient. To become weakly efficient there must not exist a point in the technology set which is better in every input and output (Charnes et al., 1996). The objective function for this model is:

$$\max \left\{ \frac{\sum_i X_i^k > 0^{\theta_i}}{\sum_i X_i^k > 0^1} \mid (\theta X^k, Y^k) \in \tau, \theta \leq \vec{1} \right\}$$

Results

The efficiencies of the security-based OSS projects were evaluated using the EMS software. The set of efficient projects identified by both distance measures was identical and listed in Table 4. Likewise, the number of inefficient projects choosing an efficient project as a benchmark was the same for all projects under both distance measures (Table 4).

Table 4 – Benchmarking of Security-based OSS projects

<i>Efficient Project Name</i>	<i># of Inefficient Projects which have chosen the Efficient Project as a Benchmark</i>
ShellTer	18
Simple Python Keylogger for Windows	17
ClamWin Free Anivirus	10
Network Security Toolkit	5
ophcrack	3
J2EE Certificate Authority, EJBCA	0
Another file integrity checker	0
BlockSSHD	0

The DEA model determines which efficient projects are used by other inefficient projects as a benchmark for efficiently transforming inputs to outputs. The model produces an index of corresponding intensities linking an inefficient project to its benchmark efficient project(s). The eight projects in Table 4 were chosen by the DEA model as “efficient benchmark projects”. These eight projects were found to be on the efficient frontier for both distance measures (Radial and MinAverage objective functions).

The DEA model is only concerned with how efficiently each project produces its outputs. The SourceForge ranking does not consider efficiency in determining project rank. This can be shown by examining the two most benchmarked projects, “ShellTer” and “Simple Python Keylogger for Windows.” These two projects were not at the top of the SourceForge ranking.

In fact, only the ‘ophcrack’ project of the top five SourceForge ranked security-based projects was identified as an efficient benchmark project by the DEA model. The two highest SourceForge ranked projects (“Open Computers and Software Inventory” and “Endian Firewall”) have 66 and 24 total bug submitters and 7 and 10 developers, respectively. These results demonstrate how the DEA model selects benchmark projects by determining how *efficiently* a project uses resources to produce outputs. The SourceForge ranking system does not consider a measure for efficiency.

A project manager can use these results to critically evaluate resources for a project and the relative efficiency of the resources. Based on the results managerial decisions on efforts to increase/decrease resources can be made and appropriate strategies developed to achieve these goals can be developed. This DEA modeling approach can be used by security-based OSS project managers to determine the relative efficacy of their project against other similar projects. These results can be used to make decisions on increasing or decreasing controllable inputs (the number of project developers) and to set goals for project outputs. Critical decisions on allotment of work effort can be made and efforts directed to more efficient projects and away from inefficient projects.

References

- Banker, R. D., Datar, S. M., and Kemerer, C. F., “A model to evaluate variables impacting the production of software Maintenance Projects”, *Management Science*, 37 (1), 1991.
- Banker, R. D. and Kemerer, C. F., “Scale Economies in New Software Development”, *IEEE Transactions on Software Development*, 15(10), 1989.
- Charnes, A., Cooper, W., and Rhodes, E., "Measuring the Efficiency of Decision Making Units", *European Journal of Operational Research* (2:6), 1978, pp 429 – 444.
- Charnes, A., Rousseau, J., and Semple, J., “Sensitivity and Stability of Efficiency Classifications in Data Envelopment Analysis”, *Journal of Productivity Analysis*, 1996, Vol. 7, 1, pp 5-18.
- Crowston, K., Annabi, H., and Howison, J., “Defining Open Source Software Project Success”, *Twenty-Fourth International Conference on Information Systems (ICIS)*, 2003, pp 327-340.
- Drakos, N., Smith, D. M., Driver, M., Natis, Y.V., Valdes, R., Pescatore, J., Weiss, G. J., Silver, M. A., Grey, M. C., Enck, J., Govekar, M., and Claunch, C., “Hype Cycle for Open-Source Technologies, 2004”, *Gartner Research*, (ID G00120900), June 2004.
- Elam, J., “Evaluating the Efficiency of IS Organizations Using Data Envelope Analysis”, *Proceedings of the International Function Point Users Group 1991 Fall Conference*, October 1991.
- Farrell, M., “The measurement of productive efficiency”, *Journal of the Royal Statistical Society, Series A*, 1957, Vol. 120, No. 3, pp 253-290.
- Greenemeier, L., “OPEN-SOURCE EXUBERANCE -- Hot-shot entrepreneurs, VC money, even paradigm shifts. Dot-coms revisited? No, it's today's flood of open-source startups”, 2005, *Information Week*, July 11.
- Hahn, J. and Zhang, C., ‘An Exploratory Study of Open Source Projects from a Project Management Perspective’, *MIS Research Workshop*, Purdue University, Fall 2005.
- Kalina, I., and Czyzycki, A., “The Ins and Outs of Open Source”, *Consulting to Management*, Vol. 16, No. 3, September 2005, pp 41-47.
- Lacy, S., “Open Source: Now It's an Ecosystem”, *BusinessWeek Online*, October 3, 2005.
- Mogull, R. and Girard, J., “Assessment Methodology for Open-Source Security Testing Tools”, *Gartner Research*, (ID G00144478), December 2006.
- Mookhey, K. K., “Open Source Tools for Security and Control Assessment”, *Information Systems Control Journal*, Vol. 1, 2004.

Paradi, J.C., Reese, D.N., and Rosen, D., "Applications of DEA to Measure the Efficiency of Software Production at Two Large Canadian Banks", *Annals of Operations Research*, 73, 1997, pg. 91-115.

Payne, C., "On the Security of Open Source Software", *Information Systems Journal*, Vol. 12, Issue 1, January 2002.

Salkever, A., "Is Open-Source Security Software Safe?" *Business Week Online*, December 12, 2001.

Scacchi, W., "Is Open Source Software Development Faster, Better and Cheaper than Software Engineering", 2nd ICSE Workshop on Open Source Software Engineering, Orlando, FL, May, 2002.

Scheel, H. "EMS: Efficiency Measurement System User's Manual", 2000. Web address: <http://www.wiso.uni-dortmund.de/lsg/or/scheel/ems/>.

Stemelos, I., Angelis, L., Oikonomou, A. and Bleris, G., "Code Quality Analysis in Open Source Software Development", *Information Systems Journal*, Vol. 12, 2002, pg. 43-60.