Chapter 7.20
Measuring the Efficiency of Free and Open Source Software Projects Using Data Envelopment Analysis

Stefan Koch
Vienna University of Economics and Business Administration, Austria

ABSTRACT

In this chapter, we propose for the first time a method to compare the efficiency of free and open source projects, based on the data envelopment analysis (DEA) methodology. DEA offers several advantages in this context, as it is a non-parametric optimization method without any need for the user to define any relations between different factors or a production function, can account for economies or diseconomies of scale, and is able to deal with multi-input, multi-output systems in which the factors have different scales. Using a data set of 43 large F/OS projects retrieved from SourceForge.net, we demonstrate the application of DEA, and show that DEA indeed is usable for comparing the efficiency of projects. We will also show additional analyses based on the results, exploring whether the inequality in work distribution within the projects, the licensing schem,e or the intended audience have an effect on their efficiency. As this is a first attempt at using this method for F/OS projects, several future research directions are possible. These include additional work on determining input and output factors, comparisons within application areas, and comparison to commercial or mixed-mode development projects.

INTRODUCTION

In the last years, free and open source software (also sometimes termed libre software) has gathered increasing interest, both from the business and academic world. As some projects in different application domains like most notably the operating system Linux together with the suite of GNU utilities, the office suites GNOME and KDE, Apache, sendmail, bind, and several programming languages have achieved huge success in their respective markets, both the adoption by commercial companies, and also the development of new business models by corporations both small and large like Netscape or IBM have increased.

Currently, any comparison of free and open source (F/OS) software projects is very difficult. There is increased discussion on how the success of F/OS projects can be defined (Stewart, 2004; Stewart and Ammeter, 2002; Crowston et al., 2004; Crowston et al., 2003), using for example search engine results as proxies (Weiss, 2005). In addition, the process applied in these projects can differ significantly.

In this paper, we propose to compare F/OS projects according to their efficiency in transforming inputs into outputs. For any production process, this efficiency and productivity is a key indicator in comparison to other processes. For F/OS projects, two levels of analysis are of interest: The F/OS process in general is different to commercial software development processes, and the process variance between F/OS projects is also high. In both cases, a main difference, and a main argument for adopting a process or elements from it, is the efficiency. Neither a commercial enterprise, nor an F/OS project would knowingly and willingly waste scarce resources by using an inefficient development process. This necessitates to compute and compare the efficiency of F/OS projects to gain an understanding of the results any process decision has on the outputs, which could lead to identifying best practices, and thus to increasing the overall efficiency and thus output of all projects.

To this end, we propose to apply the method of Data Envelopment Analysis (DEA), which is a non-parametric optimization method for efficiency comparisons without any need for the user to define any relations between different factors or a production function. In addition, DEA can account for economies or diseconomies of scale, and is able to deal with multi-input, multi-output systems in which the factors have different scales. Efficiency and productivity in software development is most often denoted by the relation of an effort measure to an output measure, using either lines-of-code (Park, 1992) or, preferably due to independence from programming language, function points (Albrecht and Gaffney, 1983). While this approach can be problematic in an environment of commercial software development as well due to missing components especially of the output, for example also Kitchenham and Mendes (2004) agree that productivity measures need to be based on multiple size measures, there are additional problems in the context of F/OS development which point towards DEA as an appropriate method.

In F/OS projects, normally the effort invested is unknown, and therefore might need to be estimated (Koch, 2004; Koch, 2005), and is also more diverse than in commercial projects, as it includes core team member, committers, bug reporters and several other groups with varying intensity of participation. Besides that, also the outputs can be more diverse. In the general case, the inputs of an F/OS project can encompass a set of metrics, especially concerned with the participants. So, in the simplest case, the number of programmers and other participants can be used. The output of a project can be measured using several software metrics like most easily the number of LOC, files, checkins to the source code control system, postings, bug reports, characteristics of development speed (e.g. coefficients of a software evolution equation estimated) or even metrics for product attributes like McCabe's cyclomatic complexity (McCabe, 1976) or object-oriented metrics, e.g. the Chidamber-Kemerer suite (Chidamber and Kemerer, 1994). This range of metrics both for inputs and outputs, and their different scales necessitates application of an appropriate method, which DEA can be.

The main result of applying DEA for a set of projects is an efficiency score for each project. This score can serve different purposes as mentioned above: First, single projects can be compared accordingly, but also groups of projects, for example those following similar process models,