Preface

There is little doubt about the profound effect the Internet has on everyday life, but the symbiotic relationship between the Internet and free and open source software (F/OSS) will be the subject of interest for many years to come. The emergence of F/OSS continues to have great impact in the way we develop, support, maintain, and distribute software. F/OSS is collaboratively built software that is shared by developers and users and can be “freely” downloaded with or without the source code for use, modification, and further distribution. Prolific licensing agreements such as the General Public License (GPL) define the rights users have over the product. F/OSS participants develop software online, relying on extensive peer collaboration through the Internet, and are motivated by a combination of intrinsic and extrinsic motives. The Bazaar, as opposed to the Cathedral, model has emerged as a viable software development approach and has produced a number of successful applications in the areas of operating systems (Linux), Web services (Apache), database applications (MySQL, PostgreSQL), and so forth. These successes may be attributed to the idiosyncratic practices adopted by F/OSS projects, which are different from the common practices in traditional software engineering projects. This new way of developing software continues to attract the research interest of many individuals and organizations.

Trends in research findings coupled with the popularity of F/OSS have highlighted a number of challenges the development model and projects face. These challenges range from issues of inactive volunteerism, quality assurance, changes in the ecology and dynamics of the communities, effective tools and practices to manage and support the software development process, the lack of explicit guidelines and specifications inherent in traditional software development, and knowledge management issues, to the technicalities of how to meet the needs of a new generation of F/OSS developers, testers, and users. F/OSS projects have adopted practices to
address some of these critical issues and help them succeed. On the other hand, a large number of projects never made it beyond the first version. However, each project is distinct and each hosting platform or portal poses unique challenges projects must surmount to succeed.

Beyond the academic and geek or hacker communities, F/OSS is making a tremendous impact in the way people work in private and public administrations and is redefining the software industry. The increased adoption of F/OSS by well-known organizations has led to more enterprise recognition and hence consideration. What is more compelling is that companies are increasingly implementing open source strategies—porting programs and applications into the Linux environment while at the same time realizing that they can charge complementary services such as post-sale services. The LAMP (Linux/Apache/MySQL/PHP or Perl) stack is one example of cost savings F/OSS offers. However, a growing number of companies take a cautionary approach towards F/OSS full adoption; the main challenges being reliability, total cost of ownership, ability to integrate with existing systems, performance, and scalability.

**Targets and Overall Objective of the Book**

Our understanding of some of the socio-technical, economical, and managerial issues surrounding F/OSS continues to increase but information on these issues remains sparse. This book leverages the expertise of authors from different parts of the world and is a compilation of software practices adopted in various F/OSS environments. Rather than being the work of one author, the book is a collection of edited chapters written by experts and practitioners in the field of software engineering, economics, sociology, mathematics, and so forth. The editors received twenty seven chapters initially, but after an extensive peer-review process, only twelve chapters were selected for inclusion in this book. The diversity in thoughts and experiences of the authors represents a joint work and cooperative action approach to F/OSS. The subject area is wide and entails many facets that the book attempts to cover:

- Empirical research on emerging F/OSS practices
- Discussion about the coordination and collaboration strategies used by various F/OSS communities
- Presentations on the tools both software users and developers use to produce quality software
- Adoption of F/OSS in public and corporate environments
- Case studies of successful and failed F/OSS projects
The book chapters combine techniques with practice to establish a synergistic view of F/OSS so that it will benefit academicians, researchers, managers, policy makers, and F/OSS developers and users at large. However, unlike its predecessors, the book does NOT begin with a concise history of F/OSS or address the ideological issue of terminology. Different terms appear in the chapters:

- Open source software (OSS)
- Free open source software (F/OSS)
- Free/libre and open source software (F/LOSS)
- Libre software

In the spirit of the domain, what F/OSS is and is not has been left to the individual authors to give their own opinion.

Sulayman K. Sowe  
Co-Editor  
Department of Informatics  
Aristotle University, Greece  
2007

## Organization of the Book

The book is organized into twelve chapters. A brief description of each of the chapters follows:

**Chapter I** presents an analysis of the evolution over time of the human resources in the Debian project and discusses, in detail, how Debian handles the volatility of the volunteers who made it happen. The authors of this chapter contend that F/OSS volunteer contributions are inherently difficult to predict, plan, and manage and yet F/OSS projects, large and small, rely on volunteers. The fairly unstructured collaboration of volunteers has demonstrated itself to be a viable software development strategy, even if it is associated with certain challenges related to project management and software quality. The reliance on volunteer contributions leads to some fundamental management challenges in most large-scale F/OSS projects.

**Chapter II** proposes for the first time a method to compare the efficiency of F/OSS projects, based on the data envelopment analysis (DEA) methodology. The author argues that DEA offers several advantages in this context, as it is a non-parametric optimization method without the need for the user to define any relations between
different factors or a production function. He further suggests that the methodology 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, the author demonstrates the application of DEA. Analysis of the results shows that DEA indeed is usable for comparing the efficiency of projects in transforming inputs into outputs, that differences in efficiency exist, and that single projects can be judged and ranked accordingly.

**Chapter III** documents the evolution and ecology of communities within the Apache project. The authors describe community structures in terms of social networks measures such as community size or degree distribution. They further explore potential growth factors and mechanisms that help explain the observed evolution of the communities. Their hypotheses about the evolution of a portfolio (or ecology) of the communities are validated through an empirical analysis of eight years of the Apache project’s mailing list archives.

**Chapter IV** presents a holistic view, and investigates the social aspects, of F/OSS communities by first analyzing existing literature on the motivation of F/OSS community participants, and then presenting the results of survey research on community participants’ perceptions of their own participation. By collecting and analyzing qualitative and quantitative data, the authors demonstrate that F/OSS communities are not just confined to support and development but are hosts to many other activities, including the promotion of F/OSS, business and personal development, and educational activities. Furthermore, it is not just source code that is openly shared but also knowledge, skills, ideas, and expertise. The authors conclude that it is the transference and dissemination of these, between a diverse set of participants, which allows F/OSS communities to function so effectively.

**Chapter V** utilizes exploratory data analysis of F/OSS projects’ documents, Web sites, mailing lists, and a review of current literature to study F/OSS communities. The chapter sets the scene for the ways in which developers and users coordinate their activities to create software. The authors discuss what F/OSS communities are and how they operate, and how organizations adopting F/OSS interact with the developers in the communities developing the software. Coordination, the authors contend, is more than getting individuals motivated to do the work they do; it also means arranging and ordering individuals’ efforts, it refers to labour division and to task specialization. Furthermore, face-to-face, semi-structured interviews with experts in The Netherlands, the U.S., and Germany are presented to better understand coordination in F/OSS communities. The authors enumerate a number of mechanisms and tools that support the processes of collaboration.

**Chapter VI** provides a starting point for discussion between the F/OSS community and end users about software quality. One important issue highlighted by the chapter is that the potential benefits associated with diversity in F/OSS come at a price, as the donated code can be of varying quality. The authors suggest that F/OSS communities need to develop appropriate mechanisms to signal quality to end users in
simple, easy to understand terms. By providing metrics for validating the quality of the F/OSS artifacts, the chapter models the overall F/OSS development process using soft systems methodology and UML. The chapter also describes how such metrics could be used to improve the development process, and demonstrate how automated extraction of such metrics could be employed alongside F/OSS repositories such as Sourceforge.net. An illustrative example is used to demonstrate the practical applicability of the proposed approach, with a description of an early prototype of the supporting information system.

Chapter VII presents a case that the future success of F/OSS is not based on the fact that there is no associated licensing fee, but on the question of the quality of the software. The authors explore the importance software quality has for software developers and discusses the conditions needed for software quality to be realized. Doing this, the authors delve into both the motivations of individual programmers and the business models of firms that sponsor F/OSS projects. While the motivations of programmers might be different from the motivations of the firms employing them, the authors argue that companies adopting F/OSS can benefit from such diverse opinions. If we understand the people and their motivations for developing F/OSS, the authors conclude that this might increase our confidence in the quality and sustainability of F/OSS and, as a result, lower the barrier to use it.

Chapter VIII seeks to identify and characterize the array of social and technical resources needed to support the development of F/OSS supporting e-commerce (EC) or e-business (EB) capabilities. The author reports on a case study within a virtual organization that has undertaken an organizational initiative to develop, deploy, and support F/OSS systems for EC or EB services. The case study of the GNUe arises from a longitudinal field study spanning four years and employing grounded theory techniques including axial coding and construction of comparative memoranda based on field data collected through face-to-face and e-mail interviews, as well as extensive collection and cross-coding of publicly available project documents and software development artifacts posted on the project’s Web site. The chapter identifies many types of socio-technical resources and resource-based capabilities for Free EC/EB that may explain/predict (a) what’s involved, (b) how it works, or (c) what conditions may shape the longer-term success or failure of such efforts. The study links F/OSS with enterprise resource planning (ERP) and EC/EB.

Chapter IX made a number of recommendations and present lessons learned from the authors’ research conducted on the migration of PAs to open source desktop software (OpenOffice.org). They report on the migration of the Government of the Brussels-Capital Region towards OpenOffice.org and integrate and compare their work to various studies on the migration of PAs towards F/OSS on the desktop. The authors present a set of best practices—based on empirical research—for the migration towards a F/OSS desktop environment.

Chapter X draws on the authors’ experience to present the technical and managerial issues SMEs and PAs should consider when introducing a GNU/Linux-based desktop
system into their corporate environment. The authors contend that for many small and medium enterprises (SMEs) and public administrations (PAs), the introduction of a GNU/Linux-based desktop system is often problematic. The technical obstacles are represented by different hardware configurations that might require several ad-hoc activities to adapt a standard GNU/Linux distribution to the specific environment. Managerial issues are related to employees’ training costs. To address these issues, the authors present DSS (Debased Scripts Set)—a next-generation live distribution that includes an unmodified Debian-based Linux release and a modular-designed file system with some extended features.

Chapter XI posits that customers who deploy technology products and/or platforms that fail to attract a thriving ecosystem or whose ecosystem deteriorates are increasingly faced with declining availability of skills, increasing operating costs, and/or lower levels of innovation. The author presents a framework and case study to highlight a number of characteristics and aspects of F/OSS projects as ecosystems. The author further characterizes individuals and organizations as members of a larger ecosystem that includes the original vendor as well as supporting foundations, external service partners, integrators, distributors, and the users themselves. The framework developed in this chapter is used to examine how such ecosystems can be evaluated by existing and potential members to gauge the health and sustainability of projects and the products and services they produce. The author’s case study is based on the open source-based Evergreen project which is sponsored by the Georgia Public Library Service (GPLS).

Chapter XII concludes the book and presents the author’s first-hand experience report about his involvement in the initiation, development, community activity coordination, and the final demise of an F/OSS project—Kalbum, which briefly flourished between October 2002 and March 2003. The story goes that, for every amazingly successful F/OSS project, there is a Kalbum somewhere, lurking in the shadows. The experience report is that of a sole developer who had to fight for time with work and family commitments working in a project which did not inherit its codebase from one of the largest, and incumbent community-developed projects in existence. If there is anything common to Kalbum and other projects, it is the F/OSS development model. The way the case study is presented is rather unusual for academic research, but the author documents the story of many untold and “unsuccessful” projects hosted at various F/OSS portals (e.g., sourceforge.net).