

Migration to FOSS: Readiness and Challenges

Huda M. A. ElHag, Hisham M. Abushama

Abstract

Free Open Source Software (FOSS) is being suggested as a panacea to a lot of computing problems, this paper addresses FOSS migration and suggests criteria for measuring migration readiness and challenges. Key success factors (KSFs) for adopting FOSS philosophy in terms of environmental Readiness and challenges to migration are presented. The paper focuses on KSFs that are thought to be critical or crucial to implement FOSS philosophy. The challenges to migration are subdivided into technical and non-technical challenges. The technical challenges include usability, software development service and support, inter-operability and integration, security and data migration, while the non-technical challenges include human factors, procurement and local language support. This paper is based on argumentative and philosophical approach. No formal research methods used in this paper; however, literature review and personal knowledge is presented.

I. INTRODUCTION

There is no doubt that the Internet has changed engineering knowledge, especially in providing rapid and often free access to information. Researchers can use the Internet to improve their professional knowledge and skills. As a result, the free open source software (FOSS) philosophy is presented. Emphasis is placed on World Wide Web search mechanisms and the opportunity that the researchers have to access information that will permit them to do their jobs better.

FOSS is a way for building, deploying, and sustaining large software systems on a global basis, and differs in many interesting ways from the principles and practices traditionally advocated for software engineering (SE) [1].

Free (as in freedom/liberty) software and open source software are often treated as the same thing [2]. However, there are differences between them with regards to the licenses assigned to the respective software. Free software generally appears licensed with the General Public License (GPL), while Open Source Software (OSS) may use either the GPL or some other license that allows for the integration of software that may not be free software [3]. Free software can be seen as a social movement [4], whereas OSS is just a software development methodology, according to free software advocates like Richard M. Stallman and the Free Software Foundation [6].

In a developing country, free and open source software can drastically cut costs both short-term and long-term, if used instead of proprietary software [27].

This paper examines both governmental use and development of software. Government enterprises today deploy both proprietary and open-source software in heterogeneous IT environments: in which proprietary and open-source software developers are use each other's software development, licensing and business models. The IT world is no longer an "either/or" world in which customers and vendors chose to be either proprietary or open source. Instead, it is an attractive world of "both/and" as the lines between proprietary and open source have been blurred, making inter-operable deployments almost inevitable in many if not most cases [28].

This paper/letter discusses key success factors (KSFs) for adopting FOSS philosophy in terms of environmental Readiness and challenges to migration. It focuses on KSFs that are thought to be critical or crucial to implement FOSS philosophy. This is based on argumentative and philosophical approach. No formal research methods used in this paper; however, literature review and personal knowledge is presented

II. FOSS READINESS KEY SUCCESS FACTORS

A. Building FOSS Community: Participants/Contributors

Community building, alliance forming, and participatory contributing are essential and recurring activities that enable FOSS projects to persist without central corporate authority. Thus, linking people, systems, and projects together through shared artifacts and sustained online discourse enables a sustained social network [4][5] and socio-technical community, Web-based information infrastructure [5], and network of alliances [4][5][6] to emerge.

This is mainly in order to form FOSS developers in one visual or "e-visual" body who are typically also end-users of the FOSS [7]. They develop and other end-users often participate in and contribute to FOSS efforts. There may also widespread recognition that FOSS projects may produce high quality and sustainable software systems that may be used by thousands to millions of end-users [7]. Subsequently, what is known about SE processes may not be equally applicable to FOSS processes without some explicit rationale or empirical justification. Thus, it is appropriate to review what is known about FOSS. However, people who participate, contribute, and join FOSS projects tend to act in ways where building trust and reputation, being creative, advancing through evermore challenging technical roles [7], as well as giving and being generous with one's time, expertise, and source code [7] are valued traits.

B. Managing FOSS Community

As mentioned in the previous section, one of the important factors in adapting FOSS philosophy is the organizing of FOSS developers in one virtual body and facilitates communication among them. Normally the tool that facilitates this is a web-site for the group. The web site may facilitate a site map that constitutes a classification of site and domain content, as well as outlining community structure and boundaries. Therefore participants can engage most frequently. Moreover, participants in FOSS projects can engage in online discussion forums or threaded email messages as a central way to observe, participate in, and contribute to public discussions of topics of interest to ongoing project participants [8].

C. Gate Keepers

Administrators of FOSS project Web sites and source code repositories also serve as gatekeepers in the choices they make for what information to post, when and where within the site to post it, as well as what not to post. Similarly, they may choose to create a site map that constitutes a classification of site and domain content, as well as outlining community structure and boundaries [9].

D. Trust and social accountability mechanisms

Developing FOSS source code and applications requires trust and accountability among project participants [10][11]. Though trust and accountability in a FOSS project may be invisible resources, ongoing software and project development work occur only when these intangible resources and mechanisms for social control are present [10][11]. These intangible resources (or "social capital") arise in many forms. They include (a) assuming ownership or responsibility of a FOSS project software module, (b) voting on the approval of individual action or contribution to ongoing project software [11], (c) shared peer reviewing [11][12], and (d) contributing gifts [3] that are reusable and modifiable common goods. They also exist through the project's recognition of a core developer's status, reputation, and geek fame. Without these attributions, FOSS developers may lack the credibility they need to bring conflicts over how best to proceed to some accommodating resolution. Finally, as a FOSS project grows in terms of the number of contributing developers, end-users, and external sponsors, then project's socio-technical mass (i.e., web of interacting resources) becomes sufficient to insure that individual trust and accountability to the project are sustained and evolving, thus enabling social networking externalities [13][12].

III. FOSS MIGRATION CHALLENGES

A. Qualification and Selection of Open Source Software

It consists of choosing metrics by which to compare the different FOSS solutions available to the organization. In this case the set of selection criteria we propose is the one standardized by the Government of Ontario IT Standard (GO-ITS)-41 [15] namely:

Functionality, Total Cost of Ownership TCO (Total cost of ownership is the cost of procuring, installing and using software, for the duration of its' life until it is retired), market share, support, maintenance, reliability, performance, scalability, usability, security, flexibility/customizability, inter-operability, Legal/license issues : As of early 2009 there are roughly 70 licenses [19] that are officially recognized as Open Source Licenses. FOSS suppliers will offer different licenses to meet the needs of different customers ranging from personal user to IT or commercial software provider. It is important to understand the various FOSS license options and to have a clear policy articulated for in an organization [20]. Typically open source applications are available "free-of-charge", but the potential for hidden costs has slowed down their acceptance in commercial settings. This licensing has an impact of open source software and the Open Source Initiative approved licenses on commercial applications, and both the legal and illegal trading of software [24].

B. Procurement

In government agencies procurement is through a call for tender, this is a very tough process for the FOSS community to participate in [17]. This is to ensure that public funds are not wasted, this does not take into account the fact that the support of a FOSS-based solution is guaranteed without any dependence on individual firms, but depends on the FOSS community, which provides all the support needed.

C. Human Factors

- 1) Reduced productivity of the personnel (caused by the new tools).
- 2) Fear of the new software
- 3) Knowledge is power [35]
- 4) Cost of training personnel for the new tools and hostility to change [16].

D. Local Development and Language Support

In a lot of countries the official language is not English and it is a challenge for the government to implement IT systems that can support these different languages, but with FOSS, that can be achieved as local developers can be used to translate different FOSS applications into local languages [34][22]. The challenge is in creating and fostering local FOSS communities. The challenges can be summarized as:

- 1) Overcoming the communication barrier between local developers and the FOSS community at large because usually information is distributed between the FAQ list, bug reports and patches.
- 2) Informing the potential local users and the public sector about the local FOSS community.
- 3) Setting up user profiles and requirements by trying to investigate the local user base.

IV. TECHNICAL CHALLENGES

A. Usability

User-centered design and other methods that in software development that produce user-friendly software are usually not used in FOSS development [36][37].

B. Software Development Service and Support

The governmental sector is increasing requiring general and customized software for public administrations [29]. If governments start using and developing FOSS applications, the message to the market is that FOSS applications are secure and ready to be widely deployed [17]. However sometimes it is not possible guarantee that development will happen. However there are two main challenges in this namely:

- What is the type of software development service required[17]. There are three main types of services
 - 1) Development from scratch
 - 2) Customizing an existing solution
 - 3) Customizing an internal existing solution
- Who provides the software development support [17] Entities providing support can be
 - 1) open source community without formal organization or legal personality
 - 2) Organized community (e.g. non profit organization of users or developers with legal personality)
 - 3) Commercial company which provides software support

C. Inter-operability and Integration

One of the motivations for the use of FOSS is to try to achieve vendor-independence, which is to retain the ability to change software products or producers in future without FOSS of data or significant FOSS of functionality [17][33]. However, this can conflict with with the requirement that the new software must be integrable with already installed operational software. "Buyers who give priority to the latter criterion instead of using a general requirement for open standards or vendor-independent inter-operability remain locked in to software they previously purchased".[17]

D. Security

Although it has been reported [30], that using FOSS does not guarantee any security benefit or limitation, security is still cited as one of the major advantages of using FOSS. The process of detecting security risks, bugs and errors in FOSS is rapid and so is the process of eliminating them, because the source code is made public [31][32][23][18]. The challenge is in obtaining metrics for measuring software security for real time and mission critical software.

E. Data Migration

Data is stored and managed by database applications. Virtually all public administrations have huge databases. Often this data is of critical importance and huge (financial) resources have been and are allocated to collect, organize, and maintain the data. [21] It is important to divide the data into categories namely [35]:

- 1) Data which can be discarded.
- 2) Data which is useful and in open format such as PDF or Postscript, or can be easily translated into open format. The cost should be considered.
- 3) Data which must be kept but which is in a legacy closed format which cannot be easily translated into an open one. This data may need copies of the legacy software.

F. FOSS Code Maintenance and Management

Software maintenance and management are time and resource consuming processes. Fault detection and correction are the main activities in the software maintenance and management. It is preferred that these activities are performed and finished in the software development process, i.e. before the software release [26]. Therefore, identifying parts of the software where testing efforts should be focused can help software engineers and project managers, in testing, inspections, and restructuring efforts towards these critical parts of the software. As a result, developers can use their resources more efficiently to deliver higher quality products in a timely manner [23] because applying equal testing and verification effort to all parts of a software system has become cost-prohibitive [25].

It is therefore important to invest in fine-grained comparison and versioning tools to carefully track changes and compare new versions of frameworks to determine the impact of upgrading to a future release [20].

V. CONCLUSION

FOSS philosophy is an alternatives approach for traditional SE processes, however yet more researches needed to study the quality of the end product of each approach, bear in mind, the similarity of the measurable attributes such as complexity, product size, resources and so forth.

FOSS research domain is growing, governments trying to free their software products from large companies' dominance and external governments' policies and rigidity. Also FOSS philosophy encourages socio-technical concept and contributes to the knowledge fairly.

The KSFs and challenges mentioned in this paper are meant to help in initiating the basic infrastructure for implementing FOSS philosophy for an environment/body wishing to implement this philosophy. However, we think there are more KSFs and challenges not covered in this letter yet to be considered here as future work for investigation.

REFERENCES

- [1] I. Sommerville, *Software Engineering*, 7th Edition, Addison- Wesley, New York, 2004.
- [2] I. P. Antoniadis, I. Samoladas, I. Stamelos, L. Angelis and G. L. Bleris, Dynamic Simulation Models of the Open Source Development Process, in [32], 174-202, 2005.
- [3] Y. Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale University Press, New Haven, CT, 2006.
- [4] M. Bergquist, and J. Ljungberg, The Power Of Gifts: Organizing Social Relationships In Open Source Communities, *Info. Systems J.*, 11, 305-320, 2001.
- [5] K. Crowston and J. Howison, Hierarchy And Centralization In Free And Open Source Software Team Communications, *Knowledge Technology & Policy*, 18(4), Winter, 65-85, 2006.
- [6] C. DiBona, D. Cooper and M. Stone, *Open Sources 2.0*, O'Reilly Media, Sebastopol, CA. 2005.
- [7] M. S. Elliott, Examining The Success of Computerization Movements in the Ubiquitous Computing Era: Free and Open Source Software Movements, in [33], 2008.
- [8] M. Elliott and W. Scacchi, Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture, in [32], 152-172, 2005.
- [9] J. Erenkrantz, Release Management within Open Source Projects, *Proceedings of 3rd. Workshop on Open Source Software Engineering*, 25th. Intern. Conf. Soft. Eng., Portland, OR, May 2003.
- [10] J. Feller, B. Fitzgerald, S. Hissam and K. Lakhani, (Eds.), *Perspectives on Free and Open Source Software*, MIT Press, Cambridge, MA, 2005.
- [11] J. Feller, B. Fitzgerald, W. Scacchi and A. Sillitti, (Eds.), *Open Source Development, Adoption and Innovation*, IFIP Vol. 234, Springer, Boston, 2007.
- [12] R. Goldman, and R. P. Gabriel, *Innovation Happens Elsewhere: Open Source as Business Strategy*, Morgan Kaufmann Publishers, San Francisco, CA, 2005.
- [13] J. Howison, M. Conklin, and K. Crowston, FLOSSmole: A Collaborative Repository for FLOSS Research Data and Analyses. *Intern. J. Info. Tech. and Web Engineering*, 1(3), 17-26, 2006.
- [14] J. West, and J. Dedrick, The Effect of Computerization Movements Upon Organizational Adoption of Open Source, in [33], 2008.
- [15] Government of Ontario IT Standard (GO-ITS)-41
- [16] P. Zuliani and G. Succi, Migrating Public Administrations To Open Source Software, *IADIS e-Society 2004 Conference*, 16-19 July, Avila, Spain. Available at "<http://www.iadis.org/es2004/>"
- [17] R. Ghosh, R. Glott and P. Schmitz, How To Work With The Public Sector: A Short Guide For Floss Developers, Report for IDABC/Open Source Observatory. Available at "<http://ec.europa.eu/idabc/en/document/2623/5585>"
- [18] S. Drozdik, G. L. Kovcs, P.Z. Kochis, Risk Assessment of an Open Source Migration Project, *OSS2005 Conference*, 11-15 July 2005, Genova, Italy. Available at "<http://oss2005.case.unibz.it>"
- [19] <http://opensource.org/licenses/alphabetical>
- [20] D. Thomas, Going Open Source Software in IT Opportunities and Challenges, *Journal Of Object Technology*, Vol. 4, No. 2, March - April 2005. Available at "<http://www.jot.fm/>"
- [21] G. L. Kovcs, Sylvester Drozdik, P. Zuliani and G. Succi, Open Source Software for the Public Administration, *Proceedings of the 6th International Workshop on Computer Science and Information Technologies CSIT2004 Budapest, Hungary*, 2004.
- [22] M. Mannila, Free And Open Source Software: Approaches In Brazil And Argentina. Tampere 2005. Available at <http://www.uta.fi/hyper/julkaisut/b/mannila/2005.pdf>
- [23] M. English, C. Exton, I. Rigon and B. Cleary, Fault Detection and Prediction in an Open-Source Software Project, *Proceedings of the 5th International Conference on Predictor Models in Software Engineering*, Vancouver, British Columbia, Canada, 2009
- [24] L. E. Cullar, Open Source License Alternatives For Software Applications: Is It A Solution To Stop Software Piracy?, *Proceedings Of The 43Rd Annual Southeast Regional Conference*, Kennesaw, Georgia, 2005.
- [25] F. Abreu and R. Carapuca, Candidate Metrics For Object-Oriented Software Within A Taxonomy Framework, *Journal of Systems and Software*, 26(1), 1994.
- [26] D. Zage and W. Zage, An Analysis of the Fault Correction Process in a Large-Scale SDL Production Model, *Proceedings of the 25th International Conference on Software Engineering*, Portland, Oregon, May, 2003.

- [27] H. Thorbergsson, T. Bjrgvinsson, . Valfells, Economic Benefits Of Free And Open Source Software In Electronic Governance, Proceedings of the 1st international conference on Theory and practice of electronic governance, Macao, China, 2007.
- [28] Stacy Avery Baird, The Heterogeneous World Of Proprietary And Open-Source Software, Proceedings of the 2nd International Conference on Theory and Practice of Electronic Governance, Cairo, Egypt, 2008.
- [29] Red Hat, (2006), Open source adoption. Available at <http://www.redhat.com/truthhappens/publicpolicy/osa>
- [30] GITOC. (2003). Using Open Source in the South African Government: A Proposed Strategy Compiled By The Government Information Technology Officers Council, available at http://www.oss.gov.za/docs/OSSstrategy_v3.pdf
- [31] Jaap-Henk Hoepman and Bart Jacobs, Increased Security Through Open Source, Communications of the ACM, v.50(1), January 2007
- [32] E. E. Kim, F/OSS Adoption in Brazil: the growth of a national strategy, (2005). Available from <http://www.blueoaxen.com/research/0000P/>
- [33] C. M. Schweik and R. C. English, Brooks' Versus Linus' Law: An Empirical Test Of Open Source Projects, Proceedings of the Association for Public Policy Analysis and Management conference, Washington, DC, USA, November 2007.
- [34] ITWEB Brainstorm, Effective e-Government, (2006).
- [35] The IDA Open Source Migration Guidelines, available from <http://www.epractice.eu/en/document/2623>
- [36] P. M. Bach, B. Kirschner and J. M. Carroll Usability And Free/Libre/Open Source Software Sig: Hci Expertise And Design Rationale, CHI '07 Extended Abstracts On Human Factors In Computing Systems, San Jose, CA, USA, 2007.
- [37] C. Benson, M. Muller-Prove and J. Mzourek Professional usability in open source projects: GNOME, OpenOffice.org, NetBeans, CHI '04 Extended Abstracts On Human Factors In Computing Systems, Vienna, Austria, 2004.