Chapter 28

User Resistance to Software Migration:
The Case on Linux

Hee-Woong Kim
Graduate School of Information, Yonsei University, Korea

Hock Chuan Chan
National University of Singapore, Singapore

So-Hyun Lee
Graduate School of Information, Yonsei University, Korea

ABSTRACT

The demand for software has increased rapidly in the global industrial environment. Open source software (OSS) has exerted significant impact on the software industry. Large amounts of resources and effort have been devoted to the development of OSS such as Linux. Based on the technology adoption model (TAM), the development of Linux as the most well-known OSS with a graphical user interface designed for ease of use and a wide range of functionalities is expected to result in high levels of Linux adoption by individual users. Linux, however, currently controls about 1% of the operating system market for personal computers. The resistance of users to switch to a new operating system remains one of the major obstacles to widespread adoption of Linux among individual users. Based on the integration of the equity implementation model and the TAM, this study examines the formation of user resistance, as well as the effects of user resistance, on the migration to Linux for personal computers. This study discusses the role and effect of user resistance based on the equity implementation model in comparison with the two main determinants in the TAM. This study contributes to the advancement of theoretical understanding of Linux migration and user resistance. The findings also offer suggestions for software communities and practitioners, of OSS in particular, to promote the use of new software by individual users.

INTRODUCTION

The demand for software has rapidly increased in today's global industrial environment. As technology environments change at an unprecedented rate, the agility of software development has become increasingly critical for software development performance (Lee & Xia, 2010; Batra, VanderMeer & Dutta, 2011). In contrast with hardware, the term software in this study refers to application software that processes the work and tasks of users. Once an individual starts to use a particular software brand, the user is often reluctant to replace it with another kind of software. Over the last 20 years, OSS products have made successful inroads into many information systems segments (Von Krogh et al., 2012). OSS has achieved great success and exerted significant impact on the software industry (Xu, Lin & Xu, 2011).

The emergence of OSS in successful projects such as Linux operating systems, Mozilla web browsers, and Apache web servers, together with the most prominent advantages of OSS, such as cost savings, freedom of modification, and availability of source codes, have created a vast interest in OSS among academics and practitioners (Ebert, 2008; Ven, Verelst & Mannaert, 2008; Li, Yan, Xu & Teo, 2011). The development and implementation of OSS has become one of the most important topics of current interest in academic, business, and political environments (Fitzgerald, 2006; Hauge, Ayala & Conradi, 2010; Singh & Tan, 2011). The OSS community and developers have been exerting significant efforts to produce software that is competitive with proprietary programs. More developers are motivated to participate in OSS software development because OSS projects are a good opportunity to improve skills and gain experience (Ke & Zhang, 2010; Roberts, Hann & Slughter, 2006). As of August 2012, Source Forge (http://sourceforge.net/), the world's largest open source development and distribution portal, was hosting over 308,000 registered projects and more than 2.7 million users.

Linux is arguably the most well known OSS project, to which large amounts of resources and effort have been devoted. Wheeler (2001) estimated that Red Hat Linux 7.1 has over 30 million physical source lines of code. Similarly, a study of Debian GNU/Linux found 300 million lines of code as of 2007 (Gonzalez-Barahona, Robles, Michlmayr, Amor & German, 2009). In comparison, Windows 98 contained an approximate 18 million source lines of code in the same time period. Using the constructive cost model (COCOMO), Red Hat Linux 7.1 is estimated to have required about 8,000 person-years of development time (Wheeler, 2001). Accordingly, if all this software had been developed through conventional proprietary means, it would have cost more than one billion U.S. dollars for development in the United States (Wheeler, 2001).

As development of OSS such as Linux has increased in general, its adoption by companies and corporate users has also increased (Ebert, 2008). The features and functionality offered by the two operating systems (i.e., Linux and Microsoft Windows) are comparable, and some users have even declared Linux better in areas such as customizability, reliability, and security (Ebert, 2008). Linux adoption and usage by individual users for personal computers, however, remains very limited. According to Market Share by Net Applications,[1] the market share for operating systems in September 2011 indicated that Linux represented 1% of the market, while Microsoft Windows had a total of 87% of the market. This creates an area of interest for study – why does Linux have a low level of individual usage despite having successfully achieved favorable comparisons to Microsoft Windows in terms of performance, usability, reliability, and functionality?

The usual determinants of the technology adoption model (TAM) (Davis, 1989) – perceived usefulness and perceived ease of use – may be able to partially explain why some people adopt the Linux software. The inclusion of a graphical user interface has enhanced Linux's ease of use

20 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/user-resistance-to-software-migration/120933

Related Content

Case Studies
Barbara Russo, Marco Scotto, Alberto Sillittiand Giancarlo Succi (2010). *Agile Technologies in Open Source Development (pp. 144-155).*
www.irma-international.org/chapter/case-studies/36502

Facilitating the Integration of Open Educational Courses
Yolanda Debose Columbus (2013). *Open-Source Technologies for Maximizing the Creation, Deployment, and Use of Digital Resources and Information (pp. 26-33).*
www.irma-international.org/chapter/facilitating-integration-open-educational-courses/70117

Predicting the Severity of Open Source Bug Reports Using Unsupervised and Supervised Techniques
Pushpalatha M Nand Mrunalini M (2021). *Research Anthology on Usage and Development of Open Source Software (pp. 676-692).*
www.irma-international.org/chapter/predicting-the-severity-of-open-source-bug-reports-using-unsupervised-and-supervised-techniques/286599

Innovation, Imitation and Open Source
Rufus Pollock (2009). *International Journal of Open Source Software and Processes (pp. 28-42).*
www.irma-international.org/article/innovation-imitation-open-source/4088

Open Source for Accounting and Enterprise Systems
Thomas Tribunellaand James Baroody (2007). *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives (pp. 555-569).*
www.irma-international.org/chapter/open-source-accounting-enterprise-systems/21216