

Chapter 4.12

Issues and Aspects of Open Source Software Usage and Adoption in the Public Sector

Gabor Laszlo
Budapest Tech, Hungary

ABSTRACT

This chapter introduces L-PEST model as the proposed tool for better understanding the fields are influenced by motivations and adaptation policy on FLOSS of public authorities and governments. Software usage in the public sector is a highly complex topic. In the confines of this chapter the selected case studies will show consideration to the vastly different needs and capacities and the different approaches and motivations towards the utilization of FLOSS by governments and/or local authorities. The primary objective of this chapter is to identify and describe the actors associated to the usage of FLOSS within and by the public sector. This chapter has made an attempt to fill this research gap and place the different actors into one complex model. It is hoped the proposed model assists better clarifying the intricate relationship between relevant factors. Nevertheless, much more research work is needed in the years

to come. According to Michel Sapin, French Minister in charge of Public Administration and e-Government (2001), “The next generation e-government has two requirements: interoperability and transparency. These are the two strengths of open source software. Therefore, I am taking little risk when I predict that open source software will take a crucial part in the development of e-Government in the years to come.”

INTRODUCTION

The digital economy transforms governments and governments took on new roles in those areas of the economy most affected by technological changes. Governments play important roles in creating the proper environment for ICT development, and also have a significant leading role as users of these technologies by creating new modes of public's behavior. Governmental functions and

operations can be managed only by the extensive use of ICTs and by using software applications (Lanvin, 2003).

The world's largest consumers of computer software are usually governments and they thus can have considerable influence on the software market. Governmental usage of software can impact on virtually all aspects of civil life: the inclusion and participation of citizens in public life, the transparency and openness of decision making, and the elimination of the digital divide, digital persistence, and digital literacy. The question of which software is utilized by public administrations is, therefore, of fundamental importance. Free Software advocate Eben Moglen has said, "Who controls the software, controls life."¹

In the early days of computing the common software model was based on the open source model. Software and hardware were often combined in a single package. The software was usually traded in the form of source code and computer users have shared their computer code. Many important early programs, also with government funding, were widely shared (Bessen, 2002).

Then, the late 1970s and early 1980s with the appearance the consumer computing saw the beginning of the commercialization of software products based on the proprietary model. The software that operates the hardware has become as important as the hardware itself.

A significant difference between open source and proprietary software is that the open source (as it is called) software source code is freely available to the user. In contrast, the proprietary software vendors release their product only in binary form and it is illegal for end users to decompile the binary machine code to usable source code.

Free/libre open source software (or FLOSS as it is commonly referred to) has gained enormous momentum all over the world. While this movement has been closely followed with attention by many advocates and practitioners, academic research on the subject has only started emerging. These research projects have focused mainly on

individual motivations, knowledge sharing and the user communities themselves.

The primary objective of this chapter is to identify and describe the factors related to the usage of open source software within and by the public sector.

To achieve this objective, background is given on the discussion about government roles and policies towards open source software, as in the selected case studies.

One of the strengths of this chapter is that it presents a theoretical framework, a general model of software usage at large within the public sector and the identified factors assigned to global perspectives.

BACKGROUND

ICTs have the capacity to play a valuable role in improving the quality of life, particularly in health, education, agriculture, and the environment. To take one example, in the healthcare sector ICTs enable the implementation of tele-health programs in remote areas, allowing some health care to be provided remotely, independent of person-to-person contact. Further, improvements in medical equipment are also a result of advances in ICTs. In education, remote access to the knowledge bases, e-libraries and even e-learning systems and universities can deliver knowledge to rural areas, where such opportunities for learning would be unavailable without ICTs. Agriculture and environmental issues can be better managed by, for example, geographic information system (GIS) and weather forecasts.

However, at the same time, there exists the so-called digital divide, an umbrella term that is commonly understood to mean the gap between ICT *haves* and *have-nots*. Generally, the approach to the question of the capacity of ICT to increase standards of living and to that of the digital divide has focused on two main issues.² One focuses mainly on actual connectivity—infrastructure

13 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/issues-aspects-open-source-software/29465

Related Content

Safety Reconfiguration of Embedded Control Systems

Atef Gharbi, Hamza Gharsellaoui, Mohamed Khalgui and Antonio Valentini (2013). *Embedded Computing Systems: Applications, Optimization, and Advanced Design* (pp. 184-210).

www.irma-international.org/chapter/safety-reconfiguration-embedded-control-systems/76957

Ontological Rules for UML-Based Conceptual Modeling: Design Considerations and a Prototype Implementation

Shan Lu and Jeffrey Parsons (2011). *International Journal of Information System Modeling and Design* (pp. 24-44).

www.irma-international.org/article/ontological-rules-uml-based-conceptual/51577

Embedding Secret Data in Digital Media Using Texture Synthesis

Suraj Krishna Patil, Prashantkumar Marutirao Gavali, Alankar Shantaram Shelar and Sandipkumar Chandrakant Sagare (2022). *International Journal of Software Innovation* (pp. 1-15).

www.irma-international.org/article/embedding-secret-data-in-digital-media-using-texture-synthesis/301225

A Lightweight Measurement of Software Security Skills, Usage and Training Needs in Agile Teams

Tosin Daniel Oyetoyan, Martin Gilje Jaatun and Daniela Soares Cruzes (2017). *International Journal of Secure Software Engineering* (pp. 1-27).

www.irma-international.org/article/a-lightweight-measurement-of-software-security-skills-usage-and-training-needs-in-agile-teams/179641

Model-Based Analysis and Engineering of Automotive Architectures with EAST-ADL

Sara Tucci-Piergiovanni, DeJiu Chen, Chokri Mraidha, Henrik Lönn, Nidhal Mahmud, Mark-Oliver Reiser, Ramin Tavakoli Kolagari, Nataliya Yakymets, Renato Librino, Sandra Torchio and Agnes Lanusse (2014). *Handbook of Research on Embedded Systems Design* (pp. 242-282).

www.irma-international.org/chapter/model-based-analysis-and-engineering-of-automotive-architectures-with-east-adl/116112