

## **The OSS Literature Gap**

The current literature of Abel (2006), Graf and Beate (2005), Faber (2002), and Kenwood (2001), presented a wealth of statistical data but failed to offer an applied model that contains raw data acquired from a pool of OSS and non-OSS institutions as well as the performance levels of OSS applications. Although Abel (2006) gave the approximate cost to implement open-source systems, there was no direction or suggestions for which configuration or implementation approach was feasible based upon an institution's current IT arrangement. There is a need for a methodology that supports managers confronted with the open-source decision-making process (Sandusky and Gasser, 2005).

Faber (2002) failed to analyze universities that were not using open-source systems in their degree programs. However, the research argued in favor of a model to derive the optimum feasibility profile of an institution. Such a system would address the performance capabilities of the architectural components, OSS applications, topologies, security, maintenance, time, expertise, fiscal requirements, and institutional IT capabilities. Abel (2006) assessed the current state of U.S. institutions by soliciting information on the cost of IT products and software. However, the findings failed to produce information that correlated to the adoption criteria and implementation techniques of OSS. By including multiple organizations in the information gathering process, the researcher will identify specific patterns to assist those universities that are exploring the possibilities of an OSS adoption.

## **Open Software vs. Closed Software**

Lin (2006) noted that proprietary and open-source applications are the two main types of software competing in today's market. In this study, both approaches will be discussed to show the standalone benefits of proprietary and OSS applications to its users. In Lin's (2006) study,

the systems were categorized as individual users or enterprise users with a need for some form of software such as an operating system or utility application. The proprietary software applications were considered as “sponsored” because the vendor controlled the property rights and intended to profit from its distribution.

OSS is a form of “unsponsored” technology meaning that there is no entry obstruction to the supply of the software; however, the distinction is that OSS does not require a supplier or a vendor but it does require expertise to deploy. In the OSS community the individual user’s expertise is simply their related knowledge and skills of the tools involved with the application. For proprietary software design at the organizational level, the user’s expertise depends on the collective knowledge and skills of the programming team assigned to the system in question. To deploy an OSS application, the cost depends on its level of expertise, meaning that the higher the expertise level, the lower the cost (Lin, 2006).

Opening the source of existing applications will initially increase their risk exposure, due to the fact that more information about system vulnerabilities will become available to hackers. However, this exposure and the associated risk of using the application can now be determined quickly and publicly. With a closed source system the perceived risks and exposures may appear to be lower than expected, while the actual exposure, due to increased familiarity of the attackers, may be significantly higher. However, because the source is open, those involved with the project can assess the exposure of a system, discover the bugs and release patches to mitigate the exposure, whereby increasing the overall security of the system. This approach will allow security fixes to be implemented quickly, so that the period of exposure is reduced (e.g., Stuart et al., 2006).

Over time, the openness of the source code will provide a platform to increase the security of the application. When programmers create sloppy code it is visible to everyone, and questions the validity and the overall quality of the application. The open-source revolution attempts to involve programmers in the process of reviewing OSS applications and implementing available tools to increase the operability of those systems. Close software applications are created and maintained by a team of dedicated developers; however, they are primarily concerned with the financial gain of the corporation not the specific needs of the consumer. In the open-source movement, new and advanced tools will be developed to improve the security of software applications. Through open-source technology, the users are afforded the ability to make a more informed choice about the security requirements of a system, based on the needs of the corporation or their independent judgment. Therefore, the literature supports the ideology that exposing the source code to the community is outweighed by the benefits of a short period of increased exposure (e.g., Stuart et al., 2006).

### **Contribution to the Literature**

This research will provide the academic community with a formal review of deployed open source applications and technological systems that are prominent in accredited higher education institutions. After the data is collected, a model and ISA program will be created to provide additional support to academic institutions that are considering the selection and implementation of OSS applications. This research will also identify the OSS vendors, licensures, and hardware technology that are utilized in higher education; however, the most significant contribution to the field will be a model and ISA application to view the current state of OSS technology and its use or lack thereof, in the higher education community.