Technological risks of Open Source Software Adoption in the organizational context – Linux in Munich (LiMux) case

Mario Silic
University of St.Gallen, Institute of Information Management,
Müller-Friedberg-Str. 8, 9000 St.Gallen, Switzerland

prof.dr. Andrea Back
University of St.Gallen, Institute of Information Management,
Müller-Friedberg-Str. 8, 9000 St.Gallen, Switzerland

March 2015

Abstract:

This article focuses on how to effectively cope with the open source software (OSS) adoption in the enterprise context. Based on the Linux in Munich case study and the identified technological risks of OSS we show challenges and risks for CIOs and propose recommendations to follow when evaluating and calculating risks of OSS adoption.

Keywords: open source, risks, CIO, Linux
1. Introduction

Back in the 80’s when IBM was nearly having the monopole in the IT world there was an axiom that many IT executives lived by when they needed to decide which technology to adopt: “No One Ever Gets Fired For Buying IBM”. While on the one side, thanks to that proven philosophy many IT careers got boosted; on the other side, many also ended for picking the wrong side. Since then things have dramatically changed. Firstly, we witnessed the disappearance of the model where software was regarded as product that we had to pay for just as we would do for the material objects (Schryen 2011). Secondly, open source movement started in the late 1970s when the GNU Project was born together with the Free Software Foundation (Stallman 1985) with objective to build a free operating system. At the same time BSD Unix was created at the University of California at Berkeley proposing one of the first open source licenses. These events reshaped and remodeled several organizational layers and proposed new economic values and strategic assets that organizations had the option to adopt or not. Today, OSS growth in enterprises is uncontestable. Two popular websites which provide web services suite and online community platform that aims to map the landscape of open source software development for hosting OSS projects, SourceForge.org and Black Duck open hub (www.openhub.net), count more than 430,000 and 660,000 projects respectively. Apache or nginx web servers are today used by 54% of all worldwide web servers (Netcraft 2014). Android (Google mobile operating system) is on the way to surpass one billion users across all devices\(^1\). Clearly, we are living an OSS era with an exponential rise of OSS products in every aspect of the end-user use. Still, one question challenges the current state of the art: why then, if OSS is seeing such an expansion, we do not see the same numbers in the enterprise context? One possible explanation is that majority of one million of OSS projects end in failures. An example from Sourceforge.org shows that only 17% of projects are really active and successful, while majority is abandoned in the early initiation stage. The explanation for this is rooted in the origins of the OSS model which was built on the foundations of freedom (freedom to run, copy, distribute, study, change and improve the software) and choice (choice to distribute your version or not) (Scacchi 2007). However, this philosophy of choice and freedom introduce risks\(^2\) that can lead to very expensive failures for organizations if not managed properly. Risk management becomes an important facet to look at as it could be a critical point to take into consideration during the OSS product implementation. Still the risk with that high number of OSS product failures remains and enterprises may be skeptical

\(^1\) http://www.gartner.com/newsroom/id/2645115

\(^2\) (Gartner 2011)
to consider products which future is uncertain and there is an overall un-clarity of the project lifecycle. Another possible explanation is that the perceived risk behind OSS is high. In 2012 a security incident affected piwik (open source web analytics application) where malicious code was injected in the OSS product that affected over 480,000 web sites. More recently, in 2014 the Heartbleed security bug found in the open source OpenSSL product affected all major web sites including Google, Facebook, Yahoo, etc. These security incidents cause high media coverage in which OSS products are usually depicted as being vulnerable, insecure and potentially dangerous for enterprises. The result of this is that many organizations are still reluctant to adopt OSS products because IT executives are unsure whether (and to what extent) they can trust OSS (Del Bianco et al. 2011; Silic 2013; Silic & Back 2013, 2015). Consequently, quite often it is not the actual security risk that impacts IT executives’ decision making processes, but rather the perceived risk that IT executives incorporate into their decision making factors regarding adoption. However, while we fully recognize the fact that proprietary software may have similar challenges, this paper is focusing solely on OSS as research setting as access to data for proprietary software is much more challenging when compared to OSS.

Based on a case study of LiMux, this article addresses the analysis of the management of the technological risks in the context of OSS by explaining and providing factors that help to reduce the perceived risk in the IT executives’ decision making process. Based on our analysis of OSS technological risk factors, we show how CIO’s and other IT decision makers can effectively cope with the OSS products adoption.

**The survey**

This study is based on the data from two different sources:

- Firstly, we obtained data from 115 CIO’s and IT decision makers by running an online survey about the importance of the technological risks. Survey was ran from June to July 2014. We asked IT decision makers to assess the technological risks by ranking them according to the importance they have during the OSS product evaluation. This importance criterion relates to the risk identification phase that decision makers perform when adopting new products. To rank technological risk factors, a scale from 0-10 was used in which 0 signified “totally irrelevant” and 10 signified “absolutely fundamental”. In order to minimize possible ranking challenges we clearly explained to participants that only the final ranking has a real meaning, while values have no meaning in themselves. In other words, giving the value of 7 to factor 1 and the value of 3 to factor 2 means that factor 1 is more important than factor 2, but the values of “7” and “3” have no meaning in themselves.
Secondly, we gathered information about LiMux case from multiple source, including: news media (Linux specialized websites such as linuxvoice.com), interviews done with LiMux project managers/stakeholders, social media (linked in website).

LiMux – Linux in Munich

LiMux project was started by the city of Munich back in 2003 after the city council took a decision to migrate from the proprietary system to open source software. Project was aiming at replacing existing Microsoft products for free and open source products. At origin of this move was the end of support for Windows NT 4.0 which the municipality was using on its more than 14,000 desktop machines. All machines were running on Windows NT 4.0 and Microsoft Office 97/2000. Other software included 300 various products such as web-browsers, schedulers and clients for Siemens BS2000 and Novell. Another complexity was related to the fact that over 170 specialization applications were also used by the municipality: macros, forms and different add-ons mainly linked to MS Office applications. In most of the cases these add-ons were developed by external companies that had contractual agreements with the city of Munich. Several other server based applications existed at the time such as (Oracle under Unix & Adabas/Natural under BS2000), file services (Novell Netware & Sun PC-Netlink), e-mail (Critical Path), calendar (Oracle), fax (Top-Call) and X.500 directory services (Critical Path). Choice that was put in front of the decision makers was to either migrate to a newer version of Microsoft products or to completely abandon the proprietary solutions for the open source. After ten years, in 2013, the project was successfully completed with the migration of 15,000 PC’s to Linux OS. This 10 year migration may be seen as a negative point and can be a road blocker for many organizations as usually the famous proverb “time is money” turns to be a tipping point for many decision makers. However, during this 10 year period many events did occur which may shed light on the technological risks related to the OSS migration and adoption in the organizational context.

Figure 1. Linux in Munich
10 year’s history

In 2013 Peter Hofmann, LiMux project leader, announced the LiMux completion: “LiMux is done, we have surpassed most of our goals and for several weeks now we have been running in regular operations mode”. Due to the high complexity of the existing environment, processes and various requirements, it was decided to go for step by step approach with migration taking place in phases. LiMux main objective was to be more independent from the software vendors and distributors (mainly Microsoft). In other words, money or the financial aspect was not the main driver of the initiative but the idea was to have more control over the upgrades of the new version and thus, be more independent. And this is exactly what OSS model was able to provide – the “freedom of choice”. Since its inception in 2011 the project went through different milestones\(^3\) that are described below:

- 2001 – 2003: Initial plans
- 2003: Official project kick off
- 2004: Official order was given to start the migration
- 2005: LiMux migration started with first PCs conversion
- 2008: Solution received certification from TÜV Germany as being user-friendly and compliant with German usability standards
- 2009: Pilot phase and start of Office migration
- 2012: 12,000 Linux-based PCs migrated
- 2013: 14,800 machines migrated to Ubuntu Linux and LibreOffice – project completed

The entire solution was built around four main components:

1. Linux Client (see Figure 2) that includes automated deployment and configuration management capabilities
2. Office software
3. WollMux - template and form manager
4. Server components

\(^3\) Detailed project milestones can be found here https://joinup.ec.europa.eu/elibrary/case/limux-it-evolution-open-source-success-story-never
Figure 2. LiMux client

Project adopted the “migration on the fly” concept (Figure 1) where all the operations should not be interrupted at any moment and normal business operations should be up and running throughout the entire migration lifecycle.

Figure 1. Migration “on the fly”

LiMux Challenges and risks

Some initial challenges relate to the fact that when the entire project was started in 2002, the OSS model was not yet that spread in the organizational context and was very much different from what OSS is today. While today, Android or Linux servers are ubiquitous, that was not the case 12 years ago. So, it is not surprising that the city of Munich, after doing the first analysis comparing the quantifiable costs of closed vs proprietary solution (Microsoft), did not find any real and quantifiable differences that could justify the OSS model. The choice that was made at the time was

---

purely based on the ideological reasons to go for the “freedom of choice” rather than staying with “vendor lock-in” solution.

Challenge #1 Compliance and Legal issues
In 2004 a legal issue appeared related to the patent situation affecting Linux where it was believed that the migration could potentially affect over 50 European patents. Those included Amazon’s “one-click” and some others related to JPEG, CIFS/SMB and XML (majority held by Microsoft). The migration was put on hold till the situation was not later clarified as posing low risk in terms of the possible patent issues.

Challenge #2 Office migration
Over 21,000 templates and 900 macros across 22 departments have been identified. Huge variety of programming languages and developers that programmed them was found in the initial project seizing. The Office migration represented one of the biggest challenges. In the end all macros and templates were consolidated into 12,000 templates and 100 macros that were centrally managed, controlled and documented.

Challenge #3 Training
One of the important challenges was related to shifting employee’s mindset from what they were habituated to something completed different: different look & feel, new templates, new processes, etc. Munich overcame this challenged by passing the right message not only to employees’ but also to all IT people that were affected by the change. And the transmitted message was about easing their every day job.

Challenge #4 Sustainability
Sustainability need to be assured as project was aiming at standardizing of the IT infrastructure which would include the consolidation of all existing documents, procedures and processes. Moreover, vendor independent release management should be guaranteed with transparency of IT costs.

Risk #1 Interoperability
One of the main identified risks related to the interoperability where exchange of documents and information could potentially cause issues. Indeed, doing such a huge migration on a large scale could bring important interoperability challenges. One of the solutions that was implemented is the implementation of the ODF (Open Document Format for
Office Applications) format that was used to edit documents. Another approach was to engage with the communication partner in case the interoperability issue arises.

Risk #2 Total Cost of Ownership (TCO)
While LiMux project clearly said that the project goal was not financially driven, project revealed that expenses for external consultants were much higher than expected. Overall, there is a debate over the TCO of the project as it was reported that the project saved tens of millions of euros to the city of Munich5, while Microsoft and HP unofficially release an unofficial study claiming that Munich would have saved €43.7 million by staying on Microsoft platform6.

Risk #3 Lack of Expertise
City of Munich had not only to hire new experienced Linux specialist but also to engage external consultants to fill the gap related to the lack of expertise. Indeed, most of it IT staff was trained on Microsoft products and important investments had to be done to acquire the new OSS skills in order to provide same level of support to end users.

Risk #4 Standardization
LiMux was fairly standardized across the different departments it took a lot of work and efforts to have the same standards at all different levels of organization. Same functionalities had to be provided to all 50 different configurations of Windows that were previously identified.

Open Source starts with users
LiMux would not be there where it without users’ acceptance and their buy-in. Passing the right message to different users can be more important than solving all technical complexities. Peter Hoffman, leader of the LiMux project explained the challenge: “There are different levels of users. Some would say: ‘This button was green before, and it isn’t green now, so I cannot work like this!’ And the others say: ‘Just give me something, I have to work, and I’ll get used to it’. We had that kind of range of users, but most were the first type.”7. Meetings and roadshows had to take place so users could see what LiMux is and how it works. Project team wanted that users get a preview of what is

5 http://www.omgubuntu.co.uk/2014/07/munich-city-saves-millions-going-open-source


7 http://www.linuxvoice.com/the-big-switch/
coming to their end and how the future will look like. He further explained: “Some people came to us and said: ‘Can I use a mouse? I thought Linux was only command line based’.

Education of users in this sense represents a must win area where OSS adoption will likely fail if users do not show and provide they clear adoption intentions.

**Which risk factors should CIO look at when adopting OSS?**

Our study of the most important technological factors for CIOs when adopting OSS product in their organizations, revealed that LiMux main risks (Interoperability, Total Cost of Ownership (TCO), Lack of Expertise and Legal) are among the top 10 study’s risk factors.

<table>
<thead>
<tr>
<th>Technological risk factor</th>
<th>Overall</th>
<th>Large</th>
<th>Medium</th>
<th>Small</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Security</td>
<td>9.6</td>
<td>10</td>
<td>9.6</td>
<td>9.6</td>
</tr>
<tr>
<td>2. Reliability</td>
<td>9</td>
<td>9.3</td>
<td>8.9</td>
<td>9</td>
</tr>
<tr>
<td>3. Performance</td>
<td>8.5</td>
<td>8.7</td>
<td>8.8</td>
<td>8.5</td>
</tr>
<tr>
<td>4. Interoperability / Maintainability</td>
<td>8.1</td>
<td>8.5</td>
<td>8.1</td>
<td>8.1</td>
</tr>
<tr>
<td>5. Lack of expertise</td>
<td>8.1</td>
<td>8.5</td>
<td>8.2</td>
<td>8.1</td>
</tr>
<tr>
<td>6. Access to the source code</td>
<td>8</td>
<td>6.5</td>
<td>7.4</td>
<td>8</td>
</tr>
<tr>
<td>7. Legal / Compliance</td>
<td>7.9</td>
<td>8.5</td>
<td>8</td>
<td>7.9</td>
</tr>
<tr>
<td>8. Availability of technical documentation/user manual</td>
<td>7.9</td>
<td>7.8</td>
<td>8.6</td>
<td>7.9</td>
</tr>
<tr>
<td>9. Total cost of ownership (TCO)</td>
<td>7.8</td>
<td>8.8</td>
<td>8</td>
<td>7.8</td>
</tr>
<tr>
<td>10. OSS quality</td>
<td>7.8</td>
<td>7</td>
<td>7.8</td>
<td>7.8</td>
</tr>
</tbody>
</table>

Table 1. Most important technological risk factors of OSS adoption for CIOs

Clearly, LiMux being a successful project offers some invaluable insights on how to adopt OSS product on larger scale with high number of dependencies and complexities that needed to be solved.

**Recommendations for OSS adoption**
Based on the most important technological risk factors identified in our survey of CIOs and on the Linux in Munich case study, we provide five recommendations to CIOs and other IT decision makers that could be helpful and valuable in their decision making process when evaluating and calculating risks related to the OSS products adoption and use in the organizational context.

1. **Support is the key**

Support is probably the most important factor in the OSS adoption. At different levels and forms (political, top level organizational, end-user, IT, etc.) without the right support OSS adoption will end in failure - just as it did in the Vienna Linux migration case. CIO should make sure that the buy-in from all the stakeholders involved is acquired as without clear and positive involvement from all interested parties failure is inevitable.

2. **Hidden costs need to be carefully calculated**

If the project will be driven purely with return on investment or total cost of ownership, the risk of failing may be high. Indeed, as it is often said: no software is completely free of charge. Despite its label where OSS is said to be “free of charge”, decision makers should be aware of all hidden costs related to the migration. One such example of hidden costs is the lack of technical knowledge on OSS product where, often, there is a need to hire an external company or get some consulting services to build the entire solution. However, despite these hidden costs, the reality is that OSS is and will remain for long very cheap in terms of the long term investment.

3. **Take time and think long-term**

Complex migrations of the existing systems and platforms to OSS will take time. In most cases, a lot of time. City of Munich case showed all the complexity of existing macros and templates, which is present in the majority of organizations. It will take time to analyze the existing and decide how to cope with the migration. Think long-term as it is the only way to follow a step-by-step migration which is inevitable in the complex organizational environments.

4. **Stay compliant and legally protected**

OSS is a jungle with huge number of the freely available products. Many are not compliant and may be using patents or pieces of code that should not be used. CIO’s should be very vigilant when it comes to the compliance and legal risks as many recent examples showed the dangers of using non-compliant OSS products. Approaches as taken by
City of Munich, where advanced legal study was done to minimize any possible risks, is one possible direction that organizations should follow.

5. Security: Check and verify OSS products

Whether it is single small OSS product or a much more complex OSS solution, every product that enters organizational ecosystem should be carefully checked and evaluated. By its definition OSS source code is available to anyone – even to malicious people who can change the source code (this does not mean that code source changes can be easily done – OSS had mechanisms in place that avoid any arbitrary change) and thus, inject potentially dangerous programs in organizational systems. Checking and validating OSS products should become part of the OSS adoption process as only by going through a rigorous and advanced testing and code check we can the guarantee that no malicious behaviors will be displayed by the OSS product.

Concluding Remarks

It is said that OSS is currently “eating the software world”\(^8\) with an exponential rise in many areas. However, in organizational context its growth is still not following the overall trend due to many risks as consequence of the open source dual use context challenges (Silic 2013; Silic & Back 2013, 2015). In this article we provided recommendations that CIOs and IT decision makers should consider when evaluating and calculating risks related to the OSS adoption.

References


