



**HAL**  
open science

# Step-by-Step Strategies and Case Studies for Embedded Software Companies to Adapt to the FOSS Ecosystem

Suhyun Kim, Jaehyun Yoo, Myunghwa Lee

► **To cite this version:**

Suhyun Kim, Jaehyun Yoo, Myunghwa Lee. Step-by-Step Strategies and Case Studies for Embedded Software Companies to Adapt to the FOSS Ecosystem. 8th International Conference on Open Source Systems (OSS), Sep 2012, Hammamet, Tunisia. pp.48-60, 10.1007/978-3-642-33442-9\_4. hal-01519083

**HAL Id: hal-01519083**

**<https://inria.hal.science/hal-01519083>**

Submitted on 5 May 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Step-by-Step Strategies and Case Studies for Embedded Software Companies to Adapt to the FOSS Ecosystem

Suhyun Kim, Jaehyun Yoo, and Myunghwa Lee  
Software Engineering Lab, Software R&D Center, Samsung Electronics  
416 Maetan-Dong, Yeongtong-Gu, Suwon, Gyeonggi-Do 443-742, Korea  
{suhyun47.kim, sjh.yoo, mhlee}@samsung.com  
WWW home page: <https://opensource.samsung.com>

**Abstract** Due to the continuous expansion of the FOSS ecosystem and the introduction of high-quality FOSS, FOSS is increasingly used in consumer electronics (CE) such as smartphones, televisions, and cameras. As a result, manufacturers of CE products have developed a close relationship with the FOSS ecosystem. For CE product manufacturers, efficient adaptation to the FOSS ecosystem has become an essential component in their business operations. This paper will divide the activities of CE product manufacturers within the FOSS ecosystem into the following four levels: identification, adoption, compliance, and contribution. It will examine the main activities and obstacles that arise in each level. The paper will also present instances where companies have succeeded in overcoming these obstacles.

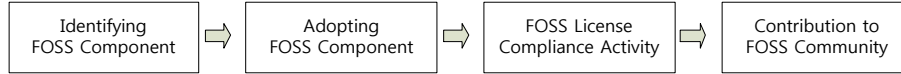
**Keywords:** FOSS, FOSS Ecosystem, Embedded Software

## 1. Introduction

FOSS initially emerged from the free software movement in response to the tendency of companies to make software their proprietary property. A software freedom activist called Richard Stallman raised objections to AT&T's policy of hiding the UNIX source code. He pioneered the concept of copyleft and introduced the GPL (General Public License) [1], which contains copyleft philosophy, while launching the GNU project [2]. The success of GPL free/open source software (FOSS) such as GNU/Linux paved the way for numerous developers to join the FOSS ecosystem [3].

As a wide range of high quality FOSS applications was introduced, many companies increased their use of FOSS to keep up with the accelerated product development cycle. The use of FOSS allows the companies to reduce development period by building a product on top of an existing FOSS application. Companies' use

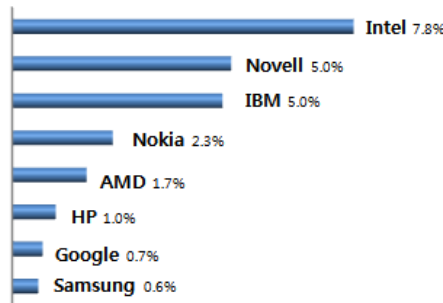
of FOSS means that they participate in the FOSS ecosystem. Fig. 1 shows the process of companies' participation in the FOSS ecosystem.



**Fig. 1.** Steps for joining the FOSS Ecosystem

Most companies tend to jump from simply identifying and using FOSS to the compliance stage. Then they reach a phase in which they modify FOSS or contribute a code that they have developed to a FOSS project.

During the early stages of participation in the FOSS-ecosystem, a company focuses on complying with the clauses of FOSS licenses, such as disclosure of the source code and acknowledgment of its use of FOSS. However, to fulfill the objective of holding a leading share of the market, as in the case of Netscape's Mozilla project [4], or to efficiently adapt to the fast-evolving FOSS environment, embedded software companies use a local patch to quickly enter the FOSS contribution stage. This patch is used by a company to maintain its own internally developed source code without applying it to the mainline of its FOSS project and apply the patch whenever a new version of FOSS is used. For your reference, Fig. 2 shows that as the use of the Linux kernel continues to increase in embedded products, companies are also increasing their contribution of Linux kernel source code [5].



**Fig. 2.** Linux kernel contribution for embedded software companies

However, there are some obstacles that companies still need to overcome before participating in the FOSS ecosystem. This paper will divide a companies' participation process in the FOSS system into the following four levels: identification, adoption, compliance, and contribution. It will then examine the main activities and obstacles that arise within each level. Examples of overcoming these obstacles will be provided. Chapter 2 will introduce the requirements for identifying

FOSS components. Chapter 3 will provide an overview of companies' activities to comply with FOSS licenses. Chapter 4 will explain the obstacles that arise in a company's contribution of FOSS. Lastly, Chapter 5 will analyze the trend of a new FOSS ecosystem that was developed in the wake of a new FOSS platform such as Android, and will introduce companies' reactions to it. Then it will draw a conclusion and propose aspects for further study.

## 2. Identification of FOSS Components

Most companies participate in the FOSS ecosystem through using FOSS. However, since there are no warranty clauses in commonly-used FOSS licenses, companies must identify and minimize the risks accompanying the use of FOSS.

In order to help select a FOSS that meets a company's requirements, many studies have been conducted to evaluate the quality of FOSS projects, such as FLOSSMetrics [6] and QualOSS [7], and to assess FOSS communities. In particular, the website ohloh.net offers a wide variety of information that allows users to access more concrete information on FOSS components, such as the number of developers and changes in the amount of source code lines for different periods of time, which can be viewed from a public repository [8].

In addition to assessing the FOSS quality and the maturity of FOSS communities provided by those studies, a company must look into the issue of copyrights/patent rights concerning FOSS. The dispute between Google and Oracle over the use of Java in Android, which is a FOSS-based operating system platform, puts a lot of pressure on Android-using manufacturers [9]. Although the FFmpeg project, which is a multimedia platform, endeavors to prevent problems involving hidden patents from emerging, it publicly maintains the FOSS philosophy-based position of no warranty. This means that if a company uses FFmpeg in its products, the company is required to pay patent royalties for all involved patents and to bear responsibility for all claims and suits filed over the neglect of paying such patent royalties [10]. This applies to all other FOSS projects.

Although there is a network called OIN [11] which shares a patent pool for open source projects and businesses to protect users from lawsuits, this kind of patent pool is rarely used in most OSS projects. Recently, lawsuits over patent rights or copyrights and court injunctions against sales have increased, but there is no "silver bullet." In this way, the use of FOSS always entails the risk of being embroiled in a lawsuit.

Some companies have established a system for examining various aspects of a newly-introduced OSS, including its patent or copyright, through the Open Source Review Board (OSRB) or others [12]. Nonetheless, such information is kept confidential within the company. Many companies do not have this kind of system and therefore, they are unable to fully examine the involvement of a third party's intellectual property in a newly-adopted OSS.

### 3. Compliance Activities

#### 3.1 GPL Violation Enforcement Organizations

GPL violation enforcement organizations came about since an increasing number of companies did not comply with FOSS licenses, with many of them failing to disclose the source code for FOSS that they used. Among the most active organizations are Europe-based `gpl-violations.org` and US-based Software Freedom Conservancy (SFC). Founded by Harald Welte, `gpl-violations.org` uses `netfilter/iptables`, `msdosfs`, and `mtd` which he copyrighted. It has filed claims against approximately 100 companies that have violated GPL licenses and won every single case [13]. As for SFC, of which Bradley Kuhn is an executive director, copyrighters of GPL-distributed Busybox gave the company the right to file lawsuits. SFC tracks down companies violating the GPL and take them to court.

In the case of `gpl-violations.org`, violators must simply comply with license clauses in order to settle the filed claims. However, SFC argues that in addition to complying with licenses, violators must also comply with all provisions of GPL if they intend to settle a legal issue involving them. When a claim is filed, it is generally processed privately between a GPL violation enforcement organization and the violating company, thereby preventing external exposure. However, when a lawsuit is filed in court, it attracts media attention regardless of the court's final ruling, thereby causing damage to the public image of the violating company. Table 1 shows the activities of violation enforcement organizations.

**Table 1.** GPL violation enforcement organizations

	<b>gpl-violations.org</b>	<b>Software Freedom Conservancy</b>
<b>Member</b>	Harald Welte	Bradley M. Kuhn
<b>Area</b>	Europe	USA
<b>Copyright Software</b>	<code>netfilter/iptables</code> , <code>msdosfs</code> , <code>mtd</code>	Busybox
<b>GPL Enforcement</b>	Claims (over 100)	Lawsuits (over 10)
<b>Restoration</b>	No need to obtain agreement with anyone. Compliance on particular GPL'ed software under ownership of <code>gpl-violations.org</code>	Need to obtain agreement with SFC. Compliance of all GPL'ed software which is contained in the product.

In addition to their independent investigation, GPL violation enforcement organizations also rely on information provided by external users to monitor companies' GPL violations. It also verifies whether or not GPL FOSS has been used by acquiring a binary of a program code contained in a product through a program called BAT (binary analysis tool). If it detects the use of the GPL-licensed FOSS, it

investigates whether the company has taken appropriate measures, such as statements of the GPL or disclosure of the source code [14].

In addition to these monitoring activities, they raise companies' awareness of this issue by informing them of common violation cases and distribute related resources such as GPL Compliance Guide to help them to comply with provisions of GPL FOSS. Companies can use these guides to check whether or not they have violated the GPL [15][16][17].

### 3.2 Compliance Activities of Companies

When a company fails to comply with the provisions in the GPL and violates a copyright law, this usually occurs because departments for developing, testing, and distributing software have a poor understanding of FOSS licenses and there is no system in place to educate them. This section will introduce various company activities designed to prevent violation of the GPL.

Companies which place importance on compliance activities provide their employees with mandatory online courses to enhance their understanding of FOSS licenses [18]. The company studied in this paper (hereinafter referred to as the "subject company") offers mandatory online courses that explain the concept and clauses of FOSS licenses, as well as proper compliance with them as shown in Fig. 3. It also offers in-class courses to delve further into this subject.

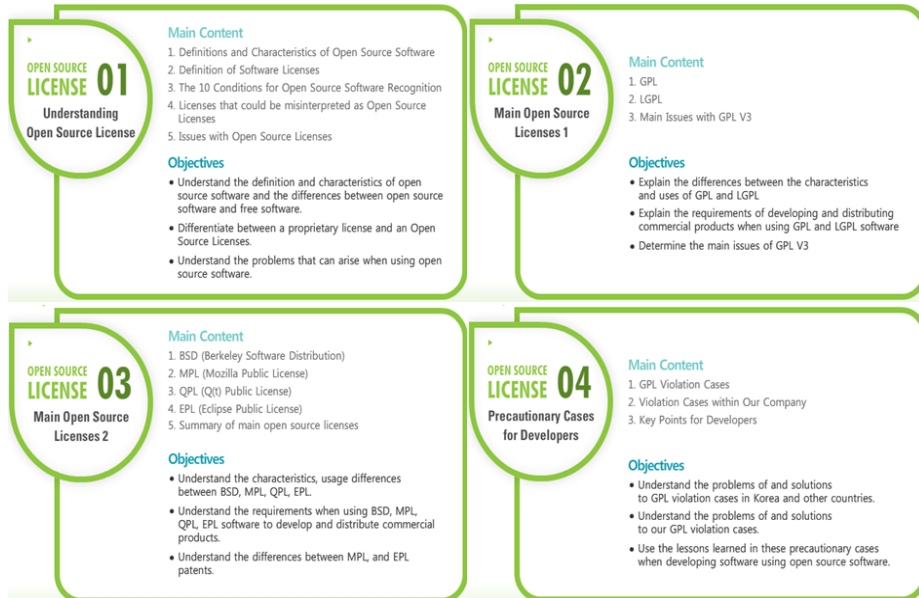


Fig. 3. Online courses on FOSS offered by the subject company

The following demonstrates how the subject company carries out activities related to FOSS compliance during the software development process. Fig. 4 shows the FOSS inspection process implemented by the companies studied.

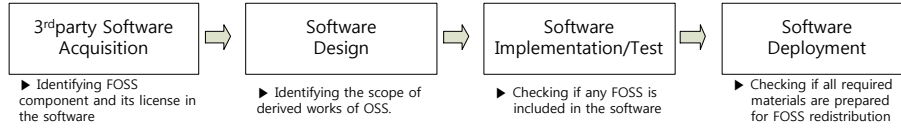


Fig. 4. FOSS inspection process of the subject company

In response to the issues introduced in Chapter 2 of this paper, the policy of checking FOSS licenses contained in software to determine whether the licenses can be used in products during the stage of adopting external software has been implemented. The subject company operates its own Open Source Advisory Board (OSAB) to handle these issues efficiently. Fig. 5 shows the composition of the OSAB. Through the OSAB, the studies and analyses of FOSS licenses and related case studies are shared and presented in written form, so that consistent OSS compliance policies can be implemented. In addition, an annual assessment is conducted to examine whether or not the verification process has been carried out in a proper manner by using a checklist similar to the Linux Foundation's Self-Assessment Compliance Checklist [19].

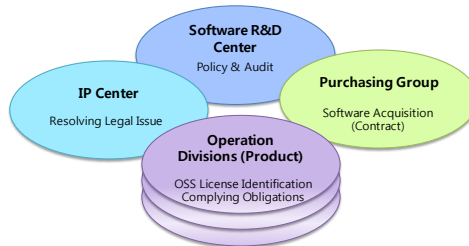


Fig. 5. The composition of OSAB for the adoption of software

It is very difficult to define the scope of GPL-derived materials [20]. In an attempt to deal with this problem, the subject company has been studying to assess the scope of GPL-derived materials during the design phase. By doing so, it prevents its core IP sector from being affected by FOSS licenses due to FOSS provisions on GPL derivative works as shown in Fig. 6.

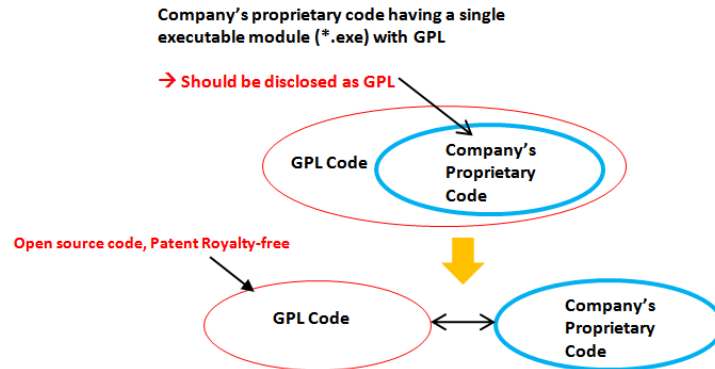


Fig. 6. Design stage reflecting the scope of GPL derivative works

During the implementation/testing phase, a code clone detection tool is used to check whether the source code includes FOSS, as shown in Fig. 7. If FOSS is detected in the source code during this process, the source code is disclosed, its use is indicated in the product manual, and the full text of the license is included in the source code during the distribution phase. This procedure has been established as an automatic system.

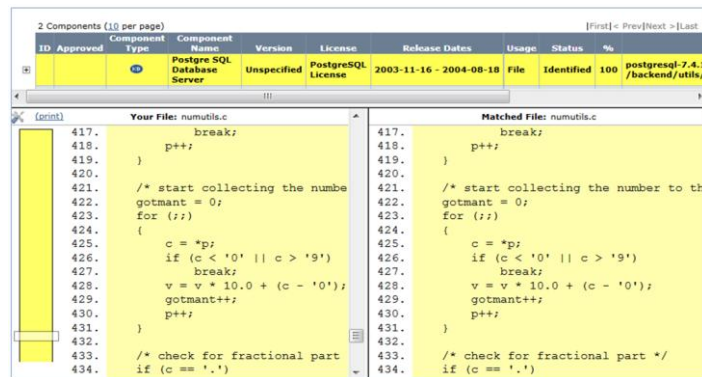


Fig. 7. Source code detection tool for FOSS

In the past, only a verification team checked the use of open sources. However, with their increasing use, the subject company has trained developers to check the use of open sources in the codes that they have developed, and to comply with clauses regarding open sources. It has also established an automated system/infrastructure to support this as shown in Fig. 8.



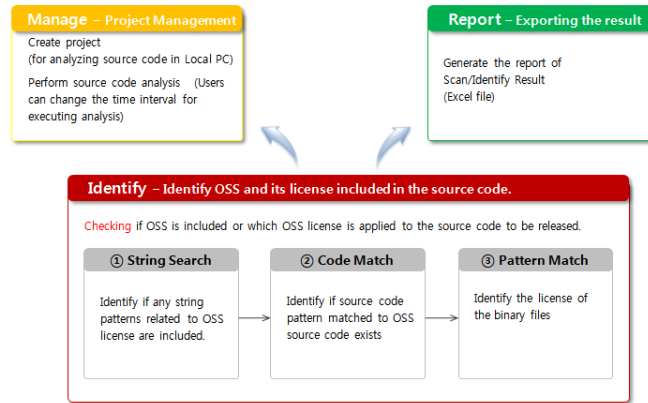


Fig. 8. FOSS self verification tool

During the distribution phase, as shown in Fig. 9, the subject company offers in writing to make FOSS source codes available on its website [21]. It also establishes a communication channel to respond to customers' inquiries on related subjects.

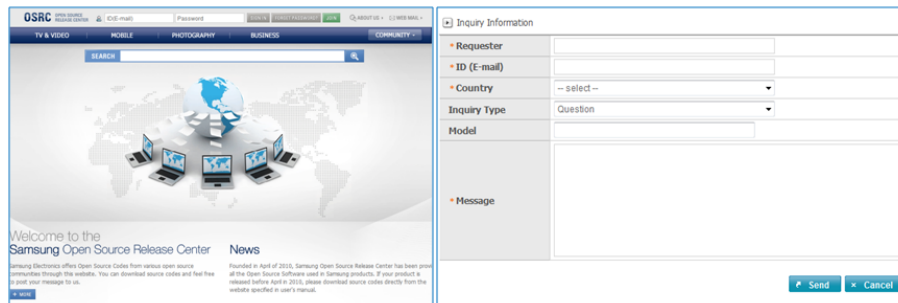


Fig. 9. A website offering source codes

#### 4. Contribution Activities

According to basic FOSS philosophy, FOSS should continue to remain FOSS (copyleft) and knowledge should be shared so that it can be developed and become even more useful [22]. If a company merely carries out the activities introduced in Chapter 3, this constitutes nothing more than complying with the license provisions on FOSS continuously remaining FOSS. This does not mean that the company has participated in the ecosystem sought by FOSS, in which knowledge is shared with others and developed further [23]. When a company minimizes local patches and remains in sync with the fast-paced evolution of both FOSS and product

development, this will benefit the company in the long term. When FOSS developers can develop source code from a company into more efficient source code, the FOSS ecosystem becomes beneficial to a greater number of users. Table 2 shows major companies' FOSS committers, as estimated by ohloh.net.

**Table 2.** FOSS committers (ohloh.net)

<b>Company</b>	<b>Committers (estimated)</b>	<b>Communities (estimated)</b>
<b>Google</b>	2,104	1,023 (Android)
<b>Apple</b>	266	Webkit
<b>Intel</b>	193	20 (Tizen)
<b>Redhat</b>	192	40 (Fedora)
<b>HP</b>	105	64 (Apache)
<b>Nokia</b>	84	28 (Symbian)

When a company undertakes such a contribution activity, it faces two major obstacles (other than the financial burdens resulting from the investment of resources in FOSS development). The first obstacle concerns the issue of protecting its IP. The second obstacle is the difficulty the company would face in becoming a FOSS project reviewer or committer, as well as possible conflict with other companies in light of the growing interest of companies in FOSS communities.

For example, in the case of the Android platform, which is released by Google under the Apache License [24], each Android maker offers its own version of Android, including unique user interface, by incorporating its own source code. Under the Apache License, however, companies usually do not provide source codes and do not apply them to the mainline of the Android Open Source Project. These files exist as local patches within companies. As a result, when Android platform source codes evolve, companies must keep up with the changes in local patches and reflect them in each of its models every single time. These local patches often contain patent applications or key technologies. Therefore, it is not easy to release source code as FOSS or make it public.

Another obstacle facing companies' contribution is an invisible wall that can block their entry into the domain. With the successful establishment of the FOSS philosophy and the increase of FOSS developers, companies try to use FOSS as part of their business strategy. In the case of WebKit, for example, a considerable number of developers who act as committers belong to Apple and Google. There is also a policy in place that requires a recommendation from existing committers or reviewers, as well as a set number of good commits. As long as this policy remains viable, it appears to be difficult for developers in competing companies to gain committer or reviewer status. Fig. 10 shows the committer ratio of companies for WebKit.

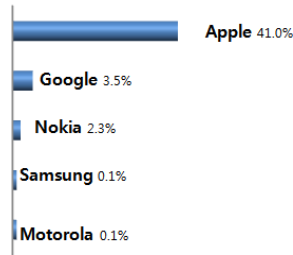


Fig. 10. Company’s committer ratio for WebKit (ohloh.net)

In order to overcome these obstacles, the subject company plans to provide its developers with various incentives that would add momentum to their FOSS contributions. It also carries out a plan to train committers through support for major communities or support programs for university students.

### 5 New FOSS Ecosystem

This chapter will introduce the product-based FOSS ecosystem which evolved from component-based ecosystems.

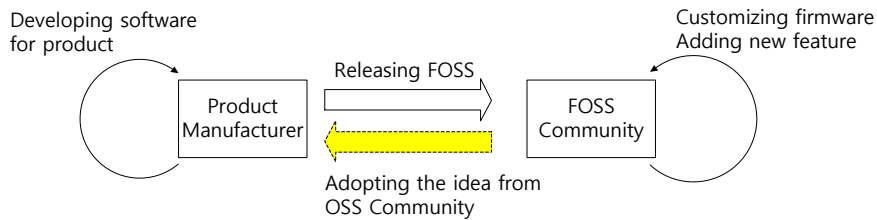


Fig. 11. Product-based FOSS ecosystem cycle

Recently, the scope of FOSS projects has expanded beyond the development of components such as Linux kernel, Webkit, and GStreamer to include platforms such as Android Open Source Project (AOSP) [ 25 ] and Tizen, which use FOSS components. In particular, companies are undertaking platform-level FOSS projects to expand potential business through the rapid dissemination of their platforms. Source codes are available on products that use open platforms. Unlike Windows Mobile, which is distributed by Microsoft, it is possible to modify them for each product. Therefore, companies are increasingly using platforms that correspond to their needs. Due to FOSS licenses, source codes of software in the products have been made available to the public. As a result, an increasing number of users have

begun to develop desired functions based on these disclosed source codes. This has generated FOSS communities which continue to grow.

**Table 3.** Activities of FOSS communities related to certain products

<b>Platform Upgrade</b>	<ul style="list-style-type: none"> <li>- GAOSP: firmware upgrade</li> <li>- GT-I9000 products: ICS upgrade</li> </ul>
<b>Performance Enhancements</b>	<ul style="list-style-type: none"> <li>- Project Voodoo: modification of file system / sound module [26]</li> <li>- Change of touch screen sensitivity</li> </ul>
<b>Usability Improvements</b>	<ul style="list-style-type: none"> <li>- Change of users' UI themes</li> <li>- CyanogenMod: development of the functions to revoke app permissions</li> <li>- Firmware and user data: development of backup/restore functions</li> </ul>

Table 3 shows the activities of FOSS communities carried out on the basis of the source codes disclosed by the companies studied. Product-based activities of FOSS communities have much more impact than component-based FOSS projects, given that the former allow developers who purchase a product to upgrade and distribute software, which can then be used by the general public. These activities, which are designed to fit with users' needs, are continuously carried out in various fields. Product-based community activities can be divided into the following three categories: Platform Upgrade, Performance Enhancements, and Usability Improvements.

Sometimes, a project may be undertaken with insufficient knowledge of the principles of the dynamics of a product and its source codes. This can result in increased after-sale service costs for manufacturing companies. In an effort to prevent this, some companies charge after-sale service fees for products that contain binaries modified by users. Some also use a lock that prevents users from making changes. However, due to community protests, they are leaning toward providing an unlock code or software. Some companies are considering adopting backup/recovery solutions for products that have been affected by the user's modification of software.

Despite these problems, the growth of product-based FOSS communities has led to another FOSS ecosystem being established. Users' practical needs are reflected in the products resulting from these communities, and this is what companies may want to pay attention to. If a company proactively responds to users' needs in their product development, it can enhance customer satisfaction. Also, users may voluntarily upgrade software in outdated products for which a company can no longer provide technical support. The number of communities and the level of their maturity toward a certain product can even affect users' decisions to purchase a product. In light of this situation, companies should disclose their source codes to support communities'

activities. The emergence of FOSS platforms has not only generated product-based FOSS, but also fueled its growth. This, in turn, has led companies to proactively participate in the ecosystem, thereby solidly establishing the product-based FOSS projects ecosystem.

## 6 Conclusion and Further Work

In this paper, a company's participation process in the more traditional component-based FOSS ecosystem was divided into four levels. The paper discussed activities and obstacles within each level and provided examples from companies that were studied. It also presented the new FOSS ecosystem which is formed from product-based FOSS communities, instead of component-based communities, following the emergence of FOSS platforms. There are still many difficulties looming ahead for companies that want to join FOSS communities, although the level of difficulty varies from one company to another. However, since participation in the FOSS ecosystem is inevitable, companies should continue to endeavor to resolve this issue. Above all, at a time when intellectual property rights are increasingly respected, companies should seek measures to minimize damages to their intellectual property rights during the distribution process of their source codes. They must also ensure that there are no intellectual property rights issues, including patent or copyright issues, when adopting FOSS.

In the future, we will continue to conduct research and activities to better adapt to the new FOSS ecosystem and to make better use of it. We are planning to monitor the activities of FOSS projects on a regular basis to discover the developer contribution field and to identify the needs of product users. In addition, we are planning to communicate with FOSS developers through the Open Source Release Center (OSRC) [27]. By proactively analyzing ideas and suggestions for improving products and then applying them, we seek to have more FOSS developers interested in our products. We also plan to make contributions to the FOSS ecosystem through various means of support for FOSS developers.

## References

- 
- [1] GNU General Public License, <http://www.gnu.org/copyleft/gpl.html>
  - [2] Stallman, R.: The GNU Manifesto, <http://www.fsf.org/gnu/manifesto.html> (1985)
  - [3] Goth, G.: Open Source Business Models: Ready for Prime Time. In: IEEE Software, November/December, pp.98-100 (2005)
  - [4] Gonzalez-Barahona, J. M., Robles, G., and Herraiz, I.: Impact of the creation of the mozilla foundation in the activity of developers. In: MSR, IEEE Computer Society, pp. 28 (2007)

- [5] Corbet, J., Kroah-Hartman, G., McPherson, A.: Linux Kernel Development How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It
- [6] Herraiz, I., Izquierdo-Cortazar, D., and Rivas-Hernandez, F.: FLOSSMetrics: Free/Libre/Open Source Software Metrics. In CSRM, IEEE Computer Society, pp. 281-284 (2009)
- [7] Deprez, J., Haaland, K. and Kamseu, F.: QualOSS Methodology and QUALOSS assessment methods. In: QualOSS project Deliverable 4.1 (2008)
- [8] Allen, J., Collison, S., and Luckey, R.: Ohloh Web Site API, <http://www.ohloh.net> (2009)
- [9] McLaughlin, L.: Inside the Software Patents Debate. In: IEEE Software, pp. 102-105 (2005)
- [10] FFmpeg License and Legal Considerations, <http://ffmpeg.org/legal.html> (2011)
- [11] Open Invention Network, <http://www.openinventionnetwork.com/>
- [12] Hewlett-Packard, FOSS Governance Fundamentals, January (2008)
- [13] Welte, H.: About the [gpl-violations.org?](http://www.gpl-violations.org/), <http://www.gpl-violations.org/about.html#why>
- [14] Hemel, A., Kalleberg, K. T., Vermaas, R., and Dolstra, E.: Finding Software License Violations through Binary Code Clone Detection. In: Proc. 8th Working Conference on Mining Software Repositories (MSR), Waikiki, Honolulu, HI, USA, pp. 63-72 (2011)
- [15] Welte, H.: GPL Product Vendor FAQ, <http://gpl-violations.org/faq/vendor-faq.html>, (2004-2010)
- [16] Hemel, A.: The GPL Compliance Engineering Guide, <http://www.loohuis-consulting.nl/downloads/compliance-manual.pdf> (2008-2010)
- [17] Schlesinger, D.: Working with Open Source: A Practical Guide. In: Interactions, ACM, pp. 35-37 (2007)
- [18] Linux Foundation, LF488 : Implementation and Management of Open Source Compliance
- [19] Linux Foundation : Self-Assessment Compliance Checklist. In: <http://www.linuxfoundation.org/programs/legal/compliance/self-assessment-checklist>
- [20] Meeker H. J.: The Open source Alternative, Understanding Risks and Leveraging Opportunities CH14 The Border Dispute of GPLs. In: The Open Source Alternative, John Wiley & Sons, Inc. (2008)
- [21] Samsung's Open Source Release Center, <http://opensource.samsung.com>
- [22] Free Software Foundation : What is Copyleft?, <http://www.gnu.org/copyleft/>
- [23] Raymond E. S.: The Cathedral and the Bazaar, O'Reilly (1999)
- [24] The Apache Software Foundation : Apache License, <http://www.apache.org/licenses/>
- [25] Open Handset Alliance.Android Open Source Project, <http://source.android.com>
- [26] Project Voodoo, <http://project-voodoo.org/>
- [27] Samsung's Open Source Community site, <https://opensource.samsung.com/community.do>