Identifying challenges for a productive and sustainable Open Source ecosystem

Jens Hardings Perl
Pontificia Universidad Católica de Chile
<jhp@ing.puc.cl>

February 23, 2007

Abstract

Open Source can be seen as an ecosystem in which all participants fulfil a role that keeps the equilibrium. When this equilibrium is altered, the ecosystem faces a challenge that can be overcome through specific countermeasures. We present a way to characterize the specific threats and identify possible countermeasures, as well as its effectiveness, using a risk analysis and handling the threat as an attack. This way, the community can be prepared to identify and face current and future challenges effectively. This document presents the methodology and some basic characterizations of some attacks, as well as the future work necessary to complete the work. The methodology proves to be useful to identify several threats and prepare the community to use its scare resources effectively to overcome current and future challenges.

1 Introduction

The participation of IT enterprises in the Open Source communities has been a big breakthrough in the last few years. The consequences of this trend are far from clear and, as the businesses models evolve, the interactions will be changing, and new challenges are likely to be emerging.

Open Source development has established itself as an ecosystem in which several participants with differing interests collaborate to achieve personal goals, achieving better results than without collaboration. While the concept of an ecosystem can be applied to several business strategies (Iansiti and Levien, 2004) and software in the general sense (Messerschmitt and Szyperski, 2003), it makes particular sense to analyze it in the context of Open Source Software (Healy and Schussman, 2003; Aigrain, 2005; Kaplan, 2005; Golden et al., 2006; Samuelson, 2006). The better results obtained by Open Source Software are due to an equilibrium or balance in the open source ecosystem. Any alteration of this equilibrium might change the rules for one or more participants, possibly making it less attractive to keep participating and thus threat the sustainability of the whole system.

As businesses have to satisfy stackholders' demand, a strategy that allows the enterprise to achieve a competitive advantage over the competition will be pursued. This is true even if that particular strategy implies unfair consequences to some of the participants, or even the possibility of breaking the open source ecosystems equilibrium which might cause a greater damage in the long term to all participants.

It makes sense to analyze how fragile the open source ecosystem is against the actions of individual or groups of participants seeking personal profit, and whether it is possible to make this ecosystem more resilient when facing the challenges posed by these individuals.

2 Proposed methodology

A good methodology for analyzing the interactions that can occur is a risk analysis similar to Cryptanalysis: identify an attack, risk or threat to the system that shows some weakness, analyze how far the attack can be taken, and propose measures to avoid or react when facing such an attack.

The attacker or source of a threat can be any person that might participate or interfere with the open source ecosystem. In practice, anybody with the right motivation to take actions. A motive can be to increase personal benefits, force a specific path to be taken, or even to terminate a project to benefit from the demand for a substitute product.

The measures used to avoid, identify or react to an attack can be of several kinds. Lessig (Lessig, 2000) identifies four dimensions in which it is possible to influence actions in the context of cyberspace:

Social even when no direct economic or legal punishment is attached to certain behavior, the condemnation of their peers is enough in many cases to avoid it.

Technological sometimes, technological measures make it difficult or even impossible to persist on an action.

Legal not conforming with legal norms makes one subject to punishment due to fines or even prison.

Economic changing the economic differences between two choices has a great impact.

Each of these measures also have its costs, so each time the community uses them, the general cost of developing software using open source methodologies also tends to increase, making the whole model slightly less appealing.

When describing an attack and possible countermeasures, they might be categorized including following aspects:

- Attacker and his objective
- Cost (negative consequences) of the attack
- Probability of the attack (Risk)
- Specific known or hypothetic cases
- Possible countermeasure, detection or avoidance
- Effectiveness of the countermeasures
- Cost of the countermeasure(s)

3 Identification of existing attacks

Even when they are not regarded as attacks, we can find several situations where the continuity of projects, and in some cases of the whole open source ecosystem has been threatened, and in many cases, countermeasures have been evaluated and implemented.

3.1 Excessive Free Riding

This is the most obvious and well-known threat to Open Source. However, free riding is also one of the main strengths of the open source model. The rules are not as rigid as to demand certain amount of retribution from each user, but the ecosystem's equilibrium demands a certain amount of participation from the overall user population.

When free riding becomes so extreme that the burden of developing the software is paid by only a few actors, they might be inclined to not continue sharing their contributions. This would be an attack that threatens the ecosystem's continuity. We can identify two different types of free riding. One is when end users of software do not participate in any way in the community, but the other is for users that also have the capability and effectively are modifying the source code and decide to not share those modifications.

3.1.1 End users free riding

These are users who would not normally engage in modifying the project, but might contribute in other ways. These include a number of possible actions, from bug reports, participation in user groups and advocacy to money donations in order to improve the project. While a typical open source project subsists with only a few users being active, the sum of those needs to fulfill a minimum threshold.

The characterization of an attack consisting of passive users is the following:

Attacker and his objective the attacker in this case would be the end users, who are only interested in getting the software and not worrying about any other aspect of the project.

Cost (negative consequences) of the attack a project with no active users will not be able to move since every effort will be on behalf of the main developers, without any help or at least feed back. The project would stagnate and be inactive, lacking any support and probably leading to "bit rot" because changes in architectures and related projects might lead to problems that need to be solved, and nobody will be around and capable of solving them efficiently.

Probability of the attack (Risk) the risk would be higher in projects that tackle specific problems and do not fit into an incremental development style. Most projects however, are said to be "scratching an itch" and thus do not fall into this category, so the probability is relatively low.

Specific known or hypothetic cases any software that solves its particular niche problem well enough not to be of interest of further improvements.

Possible countermeasure, detection or avoidance economic incentives do work well, since it keeps somebody focused on providing support. Commercial software distributions play an important role in this area. In other cases, social measures like acknowledgement

Effectiveness of the countermeasures Since the economic countermeasures depend on vendors, they might be restricted to particular configurations and leave other aspects (e.g. less popular hardware architectures) without support. The social countermeasures depend on the effective interest of the users, which is generally not too high or the problem would not exist in the first place.

Cost of the countermeasure(s) the countermeasures are self-sustained, but may require personal sacrifice which is less robust.

3.1.2 Developers/Business free riding

While it is reasonable to base a business model on open source software projects, an abuse is to take an available open source software and use it for profit while avoiding to contribute back to the community. This behavior might be within the letter of the license, but outside of its spirit.

The characterization of this attack is the following:

Attacker and his objective the attacker is a vendor that uses open source software and withholds all income, without contributing innovations, maintenance, documentation or other means back to the community.

Cost (negative consequences) of the attack the developers taking the burden of maintaining and improving the project might get tired of subsidizing third parties and decide to abandon the project.

Probability of the attack (Risk) certainly some form of this kind of free riding will be present in all successful open source projects. It is expected that free riders understand that their contribution is in their own interest and commit resources over time to contribute back. So the attack is always present but it will probably continue to be within reasonable bounds. Exceptions might be niches in which very few vendors participate, and the actions of one might have a big impact.

Specific known or hypothetic cases an enterprise dedicated to install and charge customers for a CMS developed as an open source project under the GPL license. In some cases, vendors have used such systems and even made changes to it, charging customers for the service. This is legal because the license requires to deliver source code only when a distribution of the software occurs, but not an intended consequence.

Possible countermeasure, detection or avoidance the continous analysis of business models and how they relate to the open source project's ecology will give the attackers in this case an insight and avoid the problem. Also, social pressure on behalf of the community, and economic pressure on behalf of customers will achieve great results. In the long run, changes to licenses so abusive behavior is correctly excluded in its wording will help. This is the case of some proposed changes in the transition from GPLv2 to GPLv3.

Effectiveness of the countermeasures while all of the presented countermeasures are relatively effective on its own, the best way to achieve results is to make a coordinated effort to deploy them all, in their respective time frames.

Cost of the countermeasure(s) it is necessary for someone to update the knowledge on business models related to open source, and to publicize the findings. The research community has an important role in this regard. It also requires coordination of efforts, because the various countermeasures have to be taken by different actors.

3.2 Imposition of personal agendas

In this case, the attacker might even be the leader of a software project, but it might as well be an outsider. The objective of the attacker is to impose its personal agenda over the interests of the other participants. The reasons to do so can be varied, but do not really make a difference.

We have seen many cases where projects have had problems, in which the project leaders did have important discrepancies with developers. EGCS forked from GCC in 1997 and was kept as a separate project until 1999, where both branches were merged. Emacs and its 1991 fork XEmacs are still maintained separately, due to differences in the development model. X.Org also starts as a fork from XFree86, following a series of discrepancies and a license change.

As seen in the examples, the solution in the extreme cases is to create a fork to the conflicting project, creating a competition in which the users get to decide which project they choose. However, this is a costly countermeasure and is generally used only as a last resort. The mere existence of using this possibility creates enough pressure on the project leaders to settle the problematic issues in many cases.

Certainly valid differences in engineering or legal aspects in a project may be the cause of a split into two coexisting and competing projects. However, in such cases it would be expected that both projects acknowledge each other's justification and objective differences. When this is not the case and the projects either start ignoring each other or, even worse, discrediting the other with no reasonable arguments, the confusion and frustration among users is a cost that is a consequence of this kind of attack. This cost is real even when the reasons of a split might otherwise be justified, but might be classified as another attack (see next section for an example).

The characterization of this attack is the following:

Attacker and his objective the attackers in this case are the leaders of the project. The objectives can be varied, but generally of some personal belief or whish for power or acknowledgement.

Cost (negative consequences) of the attack the cost of an unnecessary fork, users being scared away from the particular project or from open source projects in general.

Probability of the attack (Risk) for this kind of attack to materialize, the project needs to have a particularly strong leadership centered in one person, and some opposition that is just as strong and opposed.

Specific known or hypothetic cases EGS vs GCC, Emacs vs XEmacs, XFree86 vs X.org.

Possible countermeasure, detection or avoidance In this case, the main cost is also the solution to the problem: a fork.

Effectiveness of the countermeasures Due to the relatively high cost of a fork, the solution might be overly delayed, and during that time the project might stagnate.

Cost of the countermeasure(s) The cost is that the users and developers get divided into two groups, probably diminishing their weight when compared to other, possibly non-Open Source, projects.

3.3 Lack of accurate and updated information or documentation

Lack of accurate and updated information can be frustrating, just as the lack or inaccuracy of documentation. In this attack we also include the overwhelming amount of differing information from several sources.

Soluciones: una fuente de información neutral que de cuenta (overview) de la existencia de los demás proyectos. Wikipedia es un excelente candidato. Ejemplos: ghostscript.

The characterization of this attack is the following:

Attacker and his objective the attackers are the project leaders who do not deliver objective information and intend to ignore or block competing projects. In other cases, legal, technical or economical constraints make it necessary to create forks that confuse the user with too many choices.

Cost (negative consequences) of the attack users do not have the right information to choose the right project, might take the wrong choices or decide to use to non open source solutions.

Probability of the attack (Risk) in the course of competition, it can be easy to loose composure and fall into the described behavior.

Specific known or hypothetic cases the Ghostscript interpreter has several versions, including AFPL Ghostscript (formerly Alladin Ghostscript), GPL Ghostscript, GNU Ghostscript, ESP Ghostscript and Artifex (proprietary) Ghostscript. In another area, the TeX/LaTeX document system has several differing packages useful for creating presentations, forcing a user to make a through analysis before deciding which one to use.

Possible countermeasure, detection or avoidance what the user needs is accurate, neutral and trustworthy information.

Effectiveness of the countermeasures Once the user gets the necessary information from a trusted source, the problem disappears.

Cost of the countermeasure(s) The cost of creating the information is very low, but the hard part is to make it available to the user from a reliable source. Wikipedia is a good place in most cases, particularly to find the information on the Ghostscript packages.

3.4 Development Speed

As big enterprises can devote huge amounts of effort to a single project, this project might find itself overwhelmed with contributions from the people acting on behalf of the enterprise. In this case, the project might be following a path before even having the time to analyze whether that direction is right. It can be seen as a way of imposing an agenda, but where the original project is almost hijacked by certain participants.

While development speed might be a valid business strategy for an enterprise leading a project (any potential free rider has at least to be able to keep up with the development speed), it can also be a problem when the costs for the ecosystem is higher than the benefits of an increased innovation pace.

The characterization of this attack is the following:

Attacker and his objective the attacker is an enterprise that is either eager to get the project going, or intends to influence the direction the project is following. The initial leaders loose their influence on the project because of lack of resources to compete with the enterprise.

Cost (negative consequences) of the attack if the enterprise puts its interests on top of the community's, the users will have to face the problems until they can be recover the ownership of the project or create a fork, having to devote important resources to do so.

Probability of the attack (Risk) as interest of enterprises in open source continues, and as the behavior of some has been in the past, it is likely that several projects might suffer this kind of attack.

Possible countermeasure, detection or avoidance the community maintaining the project needs to enforce their rules for a fair participation of all involved members.

Effectiveness of the countermeasures while the community can impose its view on how the project will continue, it is also true that the enterprise is part of the community. It might be less clear how much this particular participant should weigh compared to others, so the effectiveness might be lower in some cases. In those cases, it might be worth to split the project when the community has two or more differing views.

Cost of the countermeasure(s) the main cost is to create clear and effective governance rules before any problems are encountered.

3.5 Forms of vendor lock-in

Customers being dependent on a particular vendor have always been a dream for the vendor benefiting from this situation, and a nightmare for the rest. While Open Source software tends to provide a solution to the vendor lock-in based on acces to source code, alternative ways to lock customers to a specific vendor are constantly being developed. Some of these alternatives can be considered an attack to the ecosystem, and may require actions. Some of these alternatives are Digital Rights Management, which are being targeted by licenses such as GPLv3.

The characterization of this attack is the following:

Attacker and his objective the attacker is an outside vendor who uses an open source software and forces by legal and/or technical means to depend on this vendor to provide future services or updates.

Cost (negative consequences) of the attack the customers loose their ability to change the provider and continue to use the software or service, making the open source software less appealing in comparison to proprietary alternatives.

Probability of the attack (Risk) the risk is high, because vendors always try to make customers return to them for new services and maintenance, locking the competition out.

Specific known or hypothetic cases one particular example is the TiVo, in which the Linux kernel is modified and used in a specific device. When the user intends to change the kernel and load it into the same device, it will not load until the vendor authorizes the modified version through a digital signature. This way, the vendor retains control over the device, leaving the customer locked into his services.

Possible countermeasure, detection or avoidance countermeasures include legal changes to licenses as to avoid the described behavior, social measures pressuring to avoid that behavior, or economic measures by boicott or other means.

Effectiveness of the countermeasures the legal measures such as license changes might be effective, but cannot be implemented in a short time span. Social measures can be effective, but the most notorious are the economical measures.

Cost of the countermeasure(s) All of the proposed changes (legal, economical and social) require big coordination efforts.

3.6 Copyleft violations

Copyleft violations are a special type of "Business free riding" attack, with the same negative impact. It is hard to identify violations and to take the necessary actions to correct the situations. In some cases, it requires buying products for reverse engineering and determining whether a violation might be happening. Also, the accusation can only be made by some author of the code, not a third party.

The characterization of this attack is the following:

Attacker and his objective the attacker is a third party who wants to get a free ride by using an open source project without complying to the license.

Cost (negative consequences) of the attack the community does not receive the contributions it is expecting. However, it is to be noted that other communitites choose not to assert a copyleft by using a license that does require it. This way, the cost of the attack is comparable to have chosen another license. For some projects this might be significant. One important point is that if no actions are taken to prevent this kind of misbehavior, the cost will increment over time and it will be ever more difficult to take action once the gpl violation has become a common practice.

Probability of the attack (Risk) as prooven by the gpl-violations.org project, 100 cases were discovered and resolved in the first two years, most of all in the embedded networking market. The risk is high, mainly due to misinformation and ignorance.

Specific known or hypothetic cases gpl-violations.org registers over 100 cases.

Possible countermeasure, detection or avoidance contacting the infringers has proven to be effective, but when this fails, legal procedures have been the alternative path.

Effectiveness of the countermeasures Considering both countermeasures they have solved 100% of the cases that have been detected so far. It is unknown how many gpl violations are continuing undetected, but we can speculate that the higher impact (and thus, cost) violations will be detected.

Cost of the countermeasure(s) the countermeasures have had a high personal cost for the leader of gpl-violations.org, considering investment in hardware, time and legal resources. Creating an organization with required funding is a useful step.

3.7 Software Patents

Software Patents have been an issue in open source licensing for a long time. They are one of the most serious threats to the open source development model, and several strategies are in place to counter it.

The characterization of this attack is the following:

Attacker and his objective vendors or other providers who see open source projects, either in general or a particular project, as a threat to their business. Also, third parties who just intend to profit from patents or promoters of patents as a way to foster innovation.

Cost (negative consequences) of the attack in the worst case scenario, open source projects might be barred from implementing particular solutions, since its licensing scheme is incompatible with patent licensing which requires a payment per copy.

Probability of the attack (Risk) the probability is very high. Software patents exist in the US and are a constant threat in Europe. The outcome of the european chapter will have a profound impact on the rest of the world.

Specific known or hypothetic cases over 30.000 software patents exist in Europe, and 16.000 new software patents are granted each year in the US.

Possible countermeasure, detection or avoidance software patents require continuous efforts, and several strategies are in place. Some intend to influcence the political decisions and barring software patents.

Others are using the very same software patents as a weapon (defensive patents) to impede third parties to enforce their own. Examples of this are the Patent Commons Project and the Open Innovation Network.

Social/economic measures are also in place, requesting enterprises working with open source communities to open up their patent porfolios.

Effectiveness of the countermeasures In europe, the work of software patent opponents has been succssful so far in the political arena. In the US it will be far more difficult because software patents are already a reality. However, if this approach is successful, it could be a definitive solution.

It is currently unknown whether projects like Open Innovation Network or Patent Commons will be successful, since their results are only measurable in the long run and once they have achieved a critical mass of patents in their portfolios. The results are more effective sooner, but they are not definitive and require constant activity to keep being effective.

Social/economic measures have proven moderately effective, with various enterprises holding many patents to offer at least a fraction of them under terms compatible with open source software. These efforts have the most immediate effect, and if they are successful, they can be used as an argument for the political efforts.

Cost of the countermeasure(s) The cost for avoiding patents on the political arena is very high, requiring the concerted effort of many people in the long run to influence the political decision makers. Similar efforts are necessary for the patent porfolio efforts and the social/economic measures, but taking into account that the results are visible sooner.

3.8 Trademark attacks

Other attacks can be the usage of trademarks, in which a project is forced to change its name.

The characterization of this attack is the following:

Attacker and his objective vendors or other providers who see open source projects, either in general or a particular project, as a threat to their business. Third parties who want to take advantage of a particular situation.

Cost (negative consequences) of the attack in the worst case scenario, open source projects might be barred from implementing particular solutions. In general the projects can be forced to change their names or cannot use the commercial and well-known names for protocols or formats they implement, as well as other annoyances to users and developers.

Probability of the attack (Risk) the probability is very high. As a matter of fact, several attacks are in place right now. A recent example is the change of the “ethereal” software to “wireshark” due to differences between the maintainer and its ex-employer who holds the trademark to “ethereal”.

Specific known or hypothetic cases Cease and desist letters have been received which threat the continuity of several projects because of trademarks, forcing name changes.

Possible countermeasure, detection or avoidance changing the name of the software project or implemented protocol generally avoids the problem.

Effectiveness of the countermeasures the name changing is generally very effective.

Cost of the countermeasure(s) The main cost a name change is the need to inform every user to avoid confusions in the future. In some cases, such as when the name of a protocol cannot be used to name an open source project, an end user may have difficulties to identify a solution because the name is less obvious. In the ethereal case, the name change was relatively painless, and information sources such as software project indexes and others are helpful to provide accurate information to users looking for a specific project under the old name.

3.9 Marketing and Information attacks

Vendors that are being threatened by open source software that competes with its products may defend themselves quite aggressively. This has happened in the past and will continue in the future. The community has been reacting to those attacks, labeled as "Fear, Uncertainty and Doubt" (FUD) speech, by providing correct and timely information. In this case, the countermeasures are both social and economic, the latter by addressing the market directly.

The characterization of this attack is the following:

Attacker and his objective the attacker are vendors and others interested in moving users away from open source projects to the software they have their interest in. The attacks can be directed towards a particular project or towards open source software in general.

Cost (negative consequences) of the attack users can be effectively driven away due to fear of using open source software.

Probability of the attack (Risk) several attacks of this sort have been identified in the past, and it is likely that they will continue to appear in the future.

Specific known or hypothetic cases general attacks have been identified against the GPL license and its copyleft characteristic, as well as against operating systems built around the Linux kernel.

Possible countermeasure, detection or avoidance it is necessary to provide users with accurate, objective and justified information to compensate the rumors and misinformation they receive. This way the users will take the right choices for the right reasons.

Effectiveness of the countermeasures the effectiveness depends directly on the quality of the information. Since open source projects and open source in general do not have marketing departments, the alliance with enterprises interested in open source is of vital importance to get the message through. But at least as important is the activity of open source community participants an researchers to provide the hard facts to back up any marketing effort in this regard.

Cost of the countermeasure(s) the personal cost of the countermeasures can be high, since many attacks require immediate response. The existance of foundations and other organizations that can react upon this sort of attack is important as the reaction does not depend on particular people.

3.10 Lock open source projects out of niches

Vendors fearing that open source might take away their business can target open source projects by locking them out of a particular niche. This lock out generally consists of a full artillery of measures, including at least legal, economic and technical aspects.

The characterization of this attack is the following:

Attacker and his objective the attacker is generally a vendor providing a complete solution for a particular niche. The attacker intends to control that niche, impeding that users can switch to other providers in general, or to open source solutions in particular.

Cost (negative consequences) of the attack open source software may be effectively barred from participating in a particular niche, and users have less choice.

Probability of the attack (Risk) the risk is high, and we can identify at least one specific case going on presently.

Specific known or hypothetic cases Digital Rights Management (DRM) is probably the best example for this kind of attack. Although it is not targeting open source, one of DRM's consequences is the impossibility of end users to use open source software to access specific data.

Possible countermeasure, detection or avoidance as well as the attack, the countermeasures need to use all of the available resources.

Effectiveness of the countermeasures this kind of attack takes a relatively long time to develop, so no hard data is available to conclude on its effectiveness.

Cost of the countermeasure(s) the countermeasures require coordination of several efforts during a long time.

4 Conclusions and future work

The proposed methodology allows the community to be more aware of current and future challenges, making it possible to devote efforts where they will have the highest impact. However, the enumeration and description of the challenges presented in this paper are far from complete and accurate, since they present only a first approach. It is necessary to backup the presented data using hard empirical data and make a more detailed analysis of these and other challenges.

According to the preliminar data presented, we can predict that in several cases the countermeasures can be very simple to implement and have predictable results with a low cost. In these cases, it is useful to avoid wasting resources in other efforts, so the analysis is valuable.

However, other challenges are more serious and require the use of all axes (social, legal, technological and economic) during an extended time period. While some of them may provide positive results in a shorter time period, it is important to not only keep on with the other axes, but to identify the best way to coordinate them in order to minimize the required effort and maximize the outcome. This coordination should also be specified in the future detailed analysis of these challenges.

References

Aigrain, P. (2005). Libre software policies at the european level. In Feller, J., Fitzgerald, B., Hissam, S., and Lakhani, K. R., editors, *Perspectives on Free and Open Source Software*, chapter 23, pages 447–459. MIT Press.

Golden, T., Larsen, S., Merling, L., Aitken, A., Fan, B., and Olson, G. (2006). Sdforum: The future of commercial open source think tank summary report. Technical report, Olliance Group.

Healy, K. and Schussman, A. (2003). The Ecology of Open-Source Software Development. http://opensource.mit.edu/papers/healyschussman.pdf.

Iansiti, M. and Levien, R. (2004). Strategy as ecology. *Harvard Business Review*.

Kaplan, J. (2005). Roadmap for open ict ecosystems. Technical report, Berkman Center for Internet & Society, Harvard Law School.

Lessig, L. (2000). *Code and Other Laws of Cyberspace*. Basic Books.

Messerschmitt, D. G. and Szyperski, C. (2003). *Software Ecosystem: Understanding an Indispensable Technology and Industry*. MIT Press.

Samuelson, P. (2006). IBM's pragmatic embrace of open source. *Communications of the ACM*, 49(10):21–25.