A New Way of Measuring Openness:
The Open Governance Index

Liz Laffan

" *How open is open enough?* "

Joel West
Professor of Innovation
and Entrepreneurship

Open source software is now "business as usual" in the mobile industry. While much attention is given to the importance of open source licenses, we argue in this article that the governance model can be as necessary to a project's success and that projects vary widely in the governance models – whether open or closed – that they employ. Open source governance models describe the control points that are used to influence open source projects with regard to access to the source code, how the source code is developed, how derivatives are created, and the community structure of the project. Governance determines who has control over the project beyond what is deemed legally necessary via the open source licenses for that project. The purpose of our research is to define and measure the governance of open source projects, in other words, the extent to which decision-making in an open source project is "open" or "closed". We analyzed eight open source projects using 13 specific governance criteria across four areas of governance: access, development, derivatives and community.

Our findings suggest that the most open platforms will be most successful in the long term, however we acknowledge exceptions to this rule. We also identify best practices that are common across these open source projects with regard to source code access, development of source code, management of derivatives, and community structure. These best practices increase the likelihood of developer use of and involvement in open source projects.

Introduction

Much has been written and debated regarding open source licenses – from the early days of the GPL license to the modern days of the Android open source platform. Yet we believe that there is one very important aspect of open source projects that has been neglected: open source governance models. While licenses determine rights to use, copy, and modify, governance determines the rights to visibility, influence, and derivative creation (Table 1). And while licenses apply to the source code, governance applies to the project or platform. More importantly, the governance model describes the control points used in an open source project – such as Android, Qt, or WebKit – and is a key determinant in the success or failure of a platform.

The governance model used by an open source project encapsulates all the hard questions about a project. Who decides on the project roadmap? How transparent are the decision-making processes? Can anyone follow the discussions and meetings taking place in the community? Can anyone create derivates based on that project? What compliance requirements are there, and how are these enforced? Governance determines who has influence and control over the project or platform – beyond what is legally deemed in the open source license. In today's world of commercially-led mobile open source projects, it is not enough to understand the open source license used by a project. It is the *governance model* that determines whether or not decision making within an open source project is open, accessible, and transparent to all users or whether it is concentrated amongst a specific set of users.

A New Way of Measuring Openness: The Open Governance Index

Liz Laffan

Table 1. Key differentiators of open source licenses versus governance

	License	Governance
Rights	Use, copy, modify	Visibility, influence, and creation of derivatives
Use	70% of projects use one of seven licenses	No agreed definition of governance
Examples	GPL, LGPL	No formal examples
Legal	Legally binding	Non-binding

Pisano and Verganti (2008; http://tinyurl.com/67bc3b) characterized open source projects as examples of an "open collaboration model" that is both open (membership) and flat (governance). Based on this view, there is an expectation that open source projects will also be governed openly. However, our findings suggest that some open source projects – such as Android, Qt, and Symbian – employ closed (hierarchical) governance models and that governance models can change over time. While Pisano and Verganti characterize governance models are either flat or hierarchical, we employ the term "open" in reference to the degree to which a project's decision-making processes are open to the community. For example, identifying who the decision makers are within open source projects (transparency) and accessing information around the actual decision-making process (accessibility) are governance criteria that are not readily captured in describing governance models as either flat or hierarchical.

In this article, we firstly explain the key governance criteria that we used to analyze eight different mobile open source projects and the outcome of this analysis. We then examine why Android has been so successful given that we find it is also the least open mobile open source project. Following from this, we identify best practices used by the most successful open source projects across the four governance areas of access, development, derivatives, and community. Finally, we suggest areas for future research and provide some conclusions regarding our research findings to date.

Analysis of Governance Models

We set out with an ambitious goal: to measure openness – the degree to which an open source project is "open" or "closed" – in ways that are rarely discussed publicly or covered in its license. We set out to define

and measure the governance of open source projects in a transparent and comprehensive manner – much like how open source licenses are defined and classified into "copyleft", "permissive", and so on. Unlike open source licenses, the governance model is made up of less visible terms, conditions, and control points that determine access, influence, decisions, and derivatives of that project.

We researched eight mobile open source projects: Android, MeeGo, Linux, Qt, WebKit, Mozilla, Eclipse, and Symbian. We selected these projects based on breadth of coverage; we picked both successful (Android) and unsuccessful projects (Symbian); both single-sponsor (Qt) and multi-sponsor projects (Eclipse); and both projects based on meritocracy (Linux) and on membership status (Eclipse).

Our research, carried out over a six-month period, included analysis of these popular open source projects and conversations with community leaders, project representatives, academics, and open source scholars. West and O'Mahony (2008; http://tinyurl.com/66fly95) identified three dimensions of open source projects: production (of source code), governance (of the open source project), and intellectual property (of the source code produced by the project). We build upon this work by also investigating how users (developers) of the project source code can influence the direction and content of the open source project through the accessibility and transparency of the decision-making processes and governance of the open source project. For example, we show how the management of source code contributions is a critical control point for governance of an open source project. Additionally, we have focused on how derivative source code (i.e., applications that can run on the open source project platform) is controlled; this is an important governance control point that is being exploited by commercial organizations supporting open source projects. Therefore, our focus has been very much on the use of the governance models as a descriptor of open source control points.

Based on our analysis, we published a report in which we proposed the Open Governance Index (OGI), a measure of open source project "openness" (Vision Mobile, 2011; http://www.visionmobile.com/research.php#OGI). The OGI comprises 13 metrics (Box 1) across the four areas of governance:

1. Access: availability of latest source code, developer support mechanisms, public roadmap, and transparency of decision making

A New Way of Measuring Openness: The Open Governance Index
Liz Laffan

2. Development: the ability of developers to influence the content and direction of the project

3. Derivatives: the ability for developers to create and distribute derivatives of the source code

4. Community: a community structure that does not discriminate between developers

The OGI quantifies how open a project is in terms of transparency, decision making, reuse, and community structure. We ranked projects across each governance parameter and on a scale of one to four on each question from Box 1. The higher the score, the more open the project. Details on how the OGI is computed, including individual scores for each project against the 13 governance criteria, are available in the full report (VisionMobile, 2011; http://www.visionmobile.com/research.php#OGI). Also note that our assessment of Qt was done before the project's governance model was revised in October 2011.

Are "Open" Projects More Successful?

A successful open source project demonstrates long-term involvement of users and developers, along with a substantial number of derivatives, and the project continually develops, matures, and evolves over time. Our research suggests that platforms that are most open will be most successful in the long-term. Eclipse, Linux, WebKit, and Mozilla each testify to this through their high OGI scores (Table 2). In terms of openness, Eclipse is by far the most open platform across access, development, derivatives, and community attributes of governance. It is closely followed by Linux and WebKit, and then Mozilla, MeeGo, Symbian, and Qt. Seven of the eight platforms reviewed fell within 30 percentage points of each other in the OGI.

Our research has identified certain attributes of successful open source projects. These attributes are: timely access to source code, strong developer tools, process transparency, accessibility to contributing code, and accessibility to becoming a committer. Equal and fair treatment of developers (i.e., "meritocracy") has become the norm and is expected by developers with regard to their involvement in open source projects.

We also note that there are common areas where most open source projects struggle to be "open". These attributes coalesce around decision making with regard to the project roadmap and committing code to the pro-

Table 2. Open Governance Index results

Project	Openness
Android	23%
Qt	58%
Symbian	58%
MeeGo	61%
Mozilla	65%
WebKit	68%
Linux	71%
Eclipse	84%

ject. In particular, we find that open source projects that originate from commercial organizations struggle most with relinquishing project control, which is not surprising, considering the structured and hierarchical decision-making structure of most organizations.

The Android paradox
Android ranks as the most closed project we examined, with an OGI score of 23%. Yet, at the same time, it is one of the most successful projects in the history of open source. Is Android proof that open governance is not needed to warrant success in an open source project?

Android's success has little to do with the open source licensing of the public codebase. Android would not have risen to its current ubiquity were it not for Google's financial muscle and famed engineering team. Development of the Android platform has occurred without the need for external developers or the involvement of a commercial community.

Google has provided Android at "less than zero" cost, since its core business is not software or search, but driving ads to eyeballs. As is now well understood, Google's strategy has been to subsidize Android such that it can deliver cheap handsets and low-cost wireless Internet access in order to drive more eyeballs to Google's ad inventory.

More importantly, Android would not have risen were it not for the billions of dollars that OEMs and network

A New Way of Measuring Openness: The Open Governance Index
Liz Laffan

Box 1. OGI Governance Criteria

Access

  1. Is source code freely available to all developers, at the same time?

  2. Is source code available under a permissive OSI-approved license?

  3. Developer support mechanisms – are project mailing lists, forums, bug-tracking databases, source code repositories, developer documentation, and developer tools available to all developers?

  4. Is the project roadmap available publicly?

  5. Transparency of decision mechanisms – are project meeting minutes/discussions publicly available such that it is possible to understand why and how decisions are made relating to the project?

Development

  6. Transparency of contributions and acceptance process – is the code contribution and acceptance process clear, with progress updates of the contribution provided (via Bugzilla or similar)?

  7. Transparency of contributions to the project – can you identify from whom source code contributions originated?

  8. Accessibility to become a committer – are the requirements and process to become a committer documented, and is this an equitable process (i.e., can all developers potentially become committers?). Note that a "committer" is a developer who can commit code to the open source project. The terms "maintainer" and "reviewer" are also used as alternatives by some projects.

  9. Transparency of committers – can you identify the committers to the project?

  10. Does the contribution license require a copyright assignment, a copyright license, or patent grant?

Derivatives

  11. Are trademarks used to control how and where the platform is used via enforcing a compliance process prior to distribution?
  12. Are go-to-market channels for applications derivatives constrained by the project in terms of approval, distribution, or discovery?

Community Structure

  13. Is the community structure flat or hierarchical (i.e., are there tiered rights depending on membership status?)

A New Way of Measuring Openness: The Open Governance Index
Liz Laffan

operators poured into Android in order to compete with Apple's iconic devices. As Stephen Elop, CEO of Nokia, said at the Open Mobile Summit in June, 2011, "Apple created the conditions necessary for Android".

However, there are some very good lessons to learn from Google's management of the Android open source project. First, Android was released as an open source project at a point in time where it was already a very advanced, complete project. OEMs, operators, and software developers could more or less immediately use it to create derivative handsets and applications. Second, Google kickstarted a developer buzz around the project with the $10 million Android Developers Challenge. Alongside financial incentives, Google sent an alluring message by opening application development within a previously inaccessible mobile industry. Finally, Google's speed of innovation (e.g., five platform versions were released in 2010) outpaces any external innovation and makes the ecosystem entirely reliant on Google.

Best Practices

Based on our research of major mobile open source projects, we have outlined the best practices for governance models. These practices are listed across the four key areas of governance: access, development, derivatives, and community.

Access
The minimum requirement for any project to be an open source project is *source-code access* such that developers can easily read, download, change, and run the code. There should be no developer discrimination; all source code should be available to all developers in a timely manner. Restrictions with regard to source code should be at a minimum, and there should be no preferential access to specific developers because this can cause friction and lead to branching of the project. All open source projects should use open source licenses that are approved by the Open Source Initiative (OSI; http://www.opensource.org).

The next most important requirement is *ease of access to developer tools, mailing lists, and forums*, such that developers can get up to speed on the specifics of the project and build and run the code with minimum effort.

Development
As much as possible, a *simple code contributions process* should operate freely and without any hindrance. While we appreciate valid intellectual property con-

cerns, such as the risk of copyright infringement, these should not complicate the contributions process any more than necessary. We also note that none of the projects reviewed in this study mandate copyright assignment; this is a good example of why copyright assignment is largely unnecessary. A broad copyright (and ideally patent) license for use of the work should suffice, provided the project has researched and identified the appropriate open source license under which to distribute the project. Copyright assignment is only ever needed when the project decides to change the terms under which it licenses the source code of the project, and this should be largely unnecessary, provided that the correct open source license is identified in the first place.

Given that the success of open source projects is largely based on the accrual of developer interest and support, we identify the *transparency of decision-making and equitable treatment of all developers* (such that they can become project committers) as being critical to long-term success. Restriction of commit rights to specific developers or organizations is a sure way to lose developer support in the long run because developers become frustrated with the inability to commit code themselves, especially if their contributions are continually rejected or ignored.

Developers often need to know where the project is headed, how it will get there, and why it is headed in that direction. They also often want the opportunity to influence the project to meet their own needs (i.e., to "scratch their own itch"). The main means by which developers can achieve this influence is by being able to commit code to the project. Therefore, it should be possible for all developers to commit code to the project, once they have shown sufficient knowledge of the code to do so. This is where *meritocracy* comes into play: those that "do" should be rewarded accordingly. Additionally the project should provide *transparent project metrics* regarding where contributions come from and who committed them.

With regard to the actual development process itself, the project should have a *policy of contribution to up-stream projects first* (if the project comprises other open source projects) such that changes and benefits accrue to up-stream and down-stream projects.

Derivatives
Compliance frameworks are becoming more and more common among open source projects in order to deter fragmentation and ensure that applications are trans-

A New Way of Measuring Openness: The Open Governance Index
Liz Laffan

ferable across multiple platforms or operating systems. However, the best mechanism to keep compliance requirements honest is to *make the compliance process as independent and transparent as possible* such that it cannot be manipulated by any one developer or organization. For example, MeeGo has asked the Linux Foundation to manage its trademark compliance requirements so that they are independent of the project.

Community
A number of the projects we reviewed use a not-for-profit foundation structure to provide independence, such that the platform is not controlled by any one organization. Other projects have established a formal association with the Linux Foundation, and this lends strong "open source credibility" to the project.

Another aspect of open source communities is the method by which authority is exercised within the community. For example, we note that both Linux and Mozilla use the benevolent dictator model, where decisions regarding disputes are made by one person. Whilst this process may work, it is still centralization of authority and decision-making, and as such it does not easily allow for others to permeate this decision-making process.

Evolving the Open Governance Index

We aim to continue the discussion on governance, to refine our criteria even further, and to make the OGI measure as meaningful as possible for the open source community. One of the first suggestions has been with regard to having a time dimension to the criteria (i.e., does openness change over time). Mature open source projects such as Eclipse, Linux, and WebKit that have stood the test of time, score quite highly with regard to openness of governance. But this has not always been the case. For example consider the following. Apple forked KHTML to create WebKit in the early 2000s, releasing the first WebKit open source project in 2005 but with reviewer and commit rights restricted to Apple personnel only which effectively sidelined the KDE community. In 2007 however Apple reversed this decision allowing allow non-Apple developers to have full commit access to the WebKit source code version control system. This shows that openness can change over a project lifecycle.

Our vision for the Open Governance Index is to for it to be a robust, and as much as is possible, an objective measure of governance for open source projects. We believe that this is necessary such that users and contribut-

ors to open source projects, including commercial entities, understand the means by which they can, or cannot, influence the direction and content of the project.

Conclusion

Today, open source software is "business as usual" in the mobile industry. It is proven that open source platforms such as Android can be as successful as proprietary platforms in terms of platform adoption, device sales, and applications development. And while open source plays a key role in developer attraction, it does not predetermine success. The mobile open source project space is undergoing consolidation to the extent that:

1. Symbian is no longer an active project, having been closed by Nokia and brought in-house while Nokia refocuses its effort using the Windows Mobile platform.

2. Nokia sold the commercial licensing rights for Qt to Digia in March 2011 and advised in November 2011 that they would "abnegate ownership" of Qt to focus on being maintainers only.

3. MeeGo is no longer being actively supported by either Nokia or Intel as an open source project, although parts of the MeeGo project are being used in the newly launched Tizen open source platform, which was launched in September 2011.

This consolidation does not detract from the fact that the mobile open source platforms can be very successful – witness Linux, Eclipse, and Android – but it does reiterate the importance of organizational support to the success of any open source project and community. To become a successful opens source project we find that there are best practices, as we have detailed in this article, which should be used to provide the best possible likelihood for success.

"Open governance" goes hand-in-hand with "open source"; it is about ensuring that developers and users have equal freedoms not to just use, but also to modify and build on the project. In many ways, open governance is the missing piece the open source licenses do not cover. Clearly, an open source license alone does not make an open project. It takes an open governance model as well. We hope our research is a step towards a fundamental change in the common understanding of how open source projects are managed and directed, including transparency regarding how decisions are made in open source projects..

A New Way of Measuring Openness: The Open Governance Index

Liz Laffan

About the Author

Liz Laffan is a Research Partner at VisionMobile. Liz has been working in the telecoms and mobile industry for over 20 years, with large telecommunications organizations, startup technology ventures, software development, and licensing firms. She holds a BA (Hons) in Business Studies from Thames Valley University and an MA in International Political Economy from Warwick University. Liz's interests lie in open source software governance and licensing and in particular how commercial organizations can best interact with open source projects.