

Empirical Study on Open Source Ecosystem

Abstract: Open source as a new type of ecosystem, facing the problem of getting more complex. Researchers keep figuring out approaches to solve the problem. Studying open source as an ecosystem is new approach. This report introduces the current research about open source ecosystems from three perspectives: open source community, open source components and making open source business sense.

1. Introduction

More and more software system companies realize that one company can rarely provide all the components, products and services. Various supply providers, standard bodies, content suppliers, service providers, and end-end organizations cooperate with each other, making the large scale software development more complicated, and even more expensive. Four decades of software engineering research have resulted in a wide range of techniques to manage the complexity of software systems development. [3]. However, the new growth of the software industries in the size and complexity drives researches to find new approaches to fulfill the task of managing the complexity.

1.1. Software Ecosystem

The term software ecosystem was introduced by Messerschmitt and Szyperski in 2003. The authors start to see the software industry as an ecosystem, introducing that software is indispensable industry, and explain software ecosystem as software and its context, including the technological and non-technological aspects. Professionals are required to deal with challenges in all stages in these chains, including analysis, design, implementing, provisioning and operation.

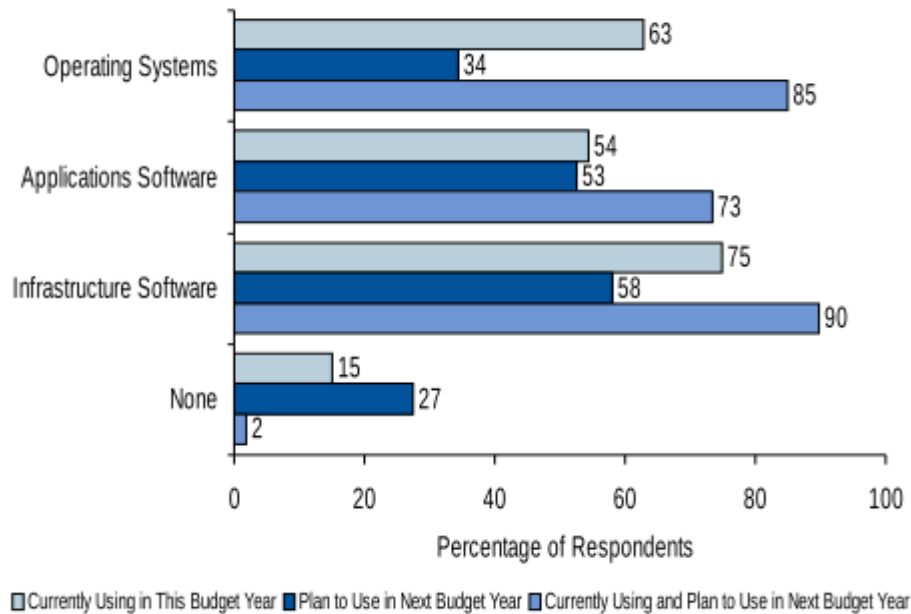
After that, there appears new studies in software ecosystem from the technological, business and social perspectives. Some of the researchers address the definitions of software ecosystem. Slinger Jansen (2009) defines a software ecosystem as “a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts. ” And for Jan Bosch(2009), software ecosystem is defined as A software ecosystem consists of the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions.

Although they are different in semantic, but at least software ecosystems include the software solutions and the transactions or relationships between them.

1.2. Open source as a mainstream

According to the Red Hat Summit 2009, open source show an exponential growth from the year of 2003 till 2007, whether measured in terms of lines of code added or new projects. And The organizer of "Open Source Think Tank" found that participants in 2007 were confident that in most enterprises, open source solutions were already in place, or would be soon. [Robert France Group, 2007]

In an investigation about whether choose open source software as the alternative of commercial software in the next budget year, the percentages of the respondents who are currently using and plan to use open source software in the next budget year are 85% for operating systems, 73% for application software and 90% for infrastructure software.



Source: Gartner 2008
 Number of respondents = 274; Multiple responses allowed.

The reasons for open source software becoming the mainstream can be divided into two aspects. One reason is that using open source software help to reduce the cost of the company especially under the context of economic crisis. Another one is that the quality assurance of open source software's performance. Negative effects of OTS components on the quality of the overall system are rare. [Jingyue Li, 2009].

2. Open source as a new type of ecosystem

RFG (2008) proposed that open source ecosystem including open communities, standards, and technologies. And it matured enough to make a business sense. So the open source ecosystem is much more about open communities, components and standards. It should be considered in a wider range. Professionals in IT or IT related business departs should also pay attention on the business value of all the elements in an ecosystem. The following content will introduce the present study from these three aspects of open source ecosystem: open community, components and business value of open source.

2.1. Open source Community

Open source is different from the closed source, since the motivations of developing software or working on an open source project are sundry, some are assigned or paid by companies, While some of them are just for their personal discretion. Moreover, the cooperation substructure of open source ecosystem is comparatively looser, and contributors may scatter around the world.

So how and why these developers determined to work on the open source projects? Then how they cooperated with each other and finally form an open source community?

2.1.1. Motivation of Software developers joining the open source projects

Software developers often start the work for a need. After a period of developing, they release a first version and draw attention of the software world and attract other developers to join them to continue the work. And also for the potential users and collaborators get a first view of this software. Different from the closed source projects, in open source, developers almost always start with a single person at their center. If they start to grow, people might volunteer to join. [1] They are not assigned to take a responsibility of one part. Instead of that, they choose what they can do according to their capabilities and then cooperated to work out the open source software. For individuals or teams, Dave Thomas (2004) concludes that: "most reasons probably come down to a combination of need, pride, ambition, or community."

2.1.2. Social networking, forming Community

In open source, developers may be interested in multiple projects and contribute to various teams, cooperating with different peoples. Also there are some relationships between open source projects, either because of the software product or for the reason of the participants in these projects.

Walt Scacchi provided an example of the networks between some of the open sources projects, where we can see forming an open source community.

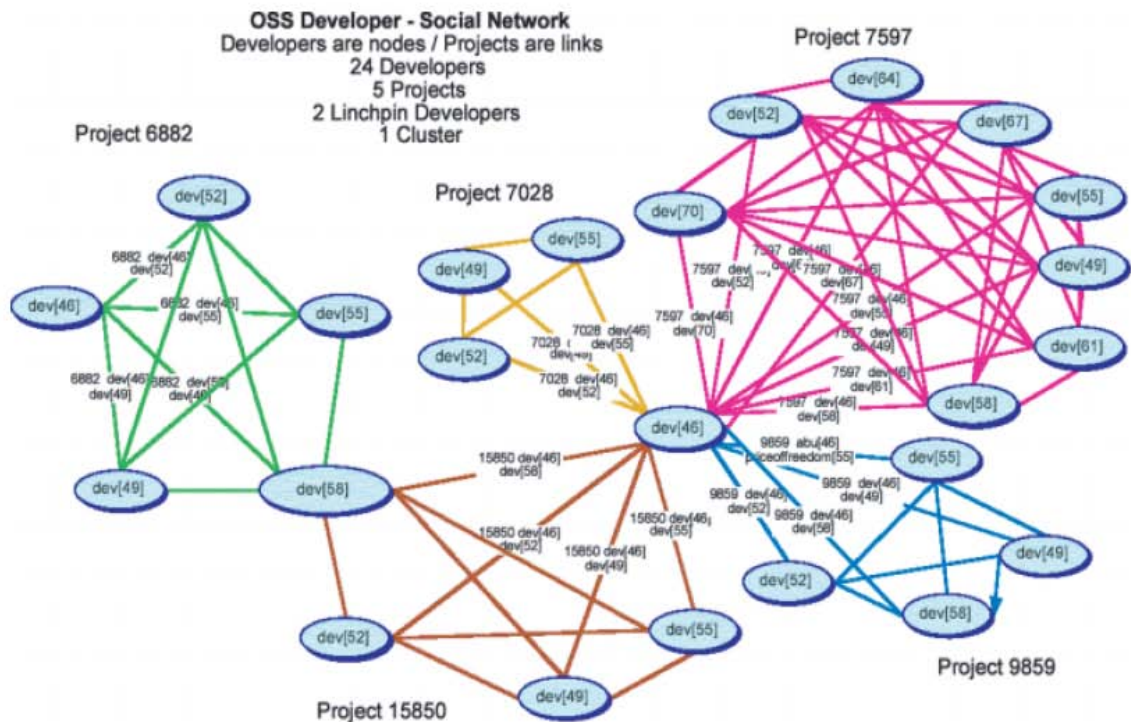


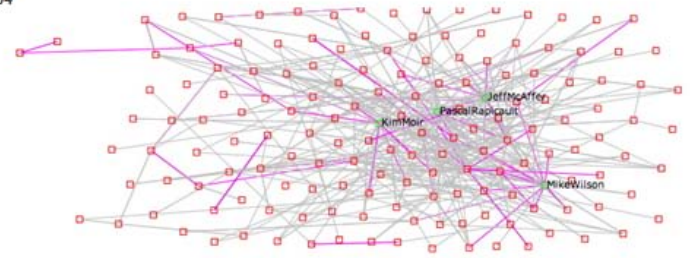
Figure 1. A social network that links 24 developers in five projects through two key developers into a larger F/OSS project community (Madye *et al.* 2004)

And there are the networks of the eclipse word wide discussion group on one website, recording how the networks start and evolve from the year 2001 to the year 2006. They also concluded from the evolution that activities decayed since 2003 due to the better cater of the developers needs when working on an new open source project. [7].

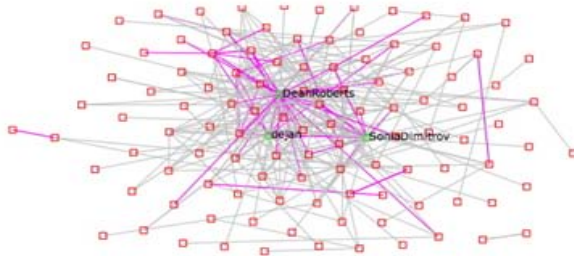
2001



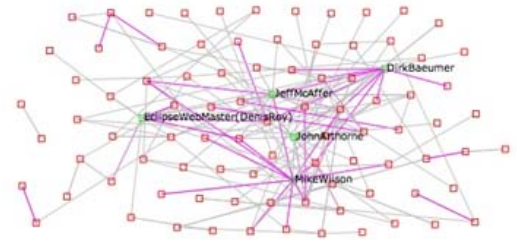
2004



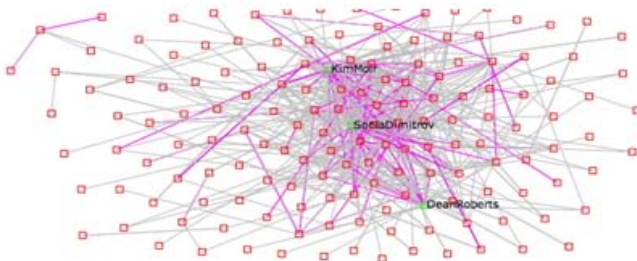
2002



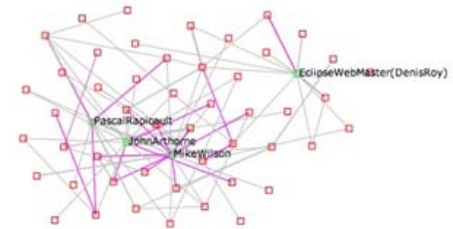
2005



2003



2006



2.2. Open source Components

There are some researches focusing on the technological side of the open source ecosystem. Since various open sourcing resources scatter globally and professionals have the freedom of choosing open source systems, the integration of open source components turns complicated. And every open source system has its own technological architecture, and it is difficult to draw up what components should be inside generally. Gunnar Stevens (2010) looks into the open source eclipse ecosystem and provide practices of appropriating eclipse ecosystems, that may help the technological integration work of open source systems on the eclipse platform.

One of the most important of Gunnar Steven emphasizes is taking the perspective of users. Tow practice about component network from the aspect of users are: "Design for Flexibility; Managing Flexibility." He addresses Eclipse and Mozilla Firefox as representative of new type of

open software that “can be supplemented by manifold extensions, being implemented by independent software vendors and open source projects.”Eclipse is an evolving socio-technical network, where technical dependencies between individual plug-ins are negotiated between different actors in the environment of related socio-economic dependencies. (Gunnar Steven, 2010)

Zhihui Yang and Michael Jiang (2007) summarize a comparison among eclipse and other integration platform as follows:

Table I

Tool-integration platform comparison

	XML-based tools	Eclipse	NetBeans	IntelliJ IDEA
Java	N/A	Yes	Yes	Yes
C/C++	N/A	Yes	Yes	No
Project/component integration	No	Yes	No	No
Data integration	Yes	Yes	Yes	Yes
Function integration	Yes	Yes	Yes	Yes
User interface integration	No	Yes	Yes	Yes
DSM/DSL support*	No	Yes	Yes	No
Available plug-ins†	N/A	1,385	71	358
Human languages	N/A	23	1 (English)	1 (English)

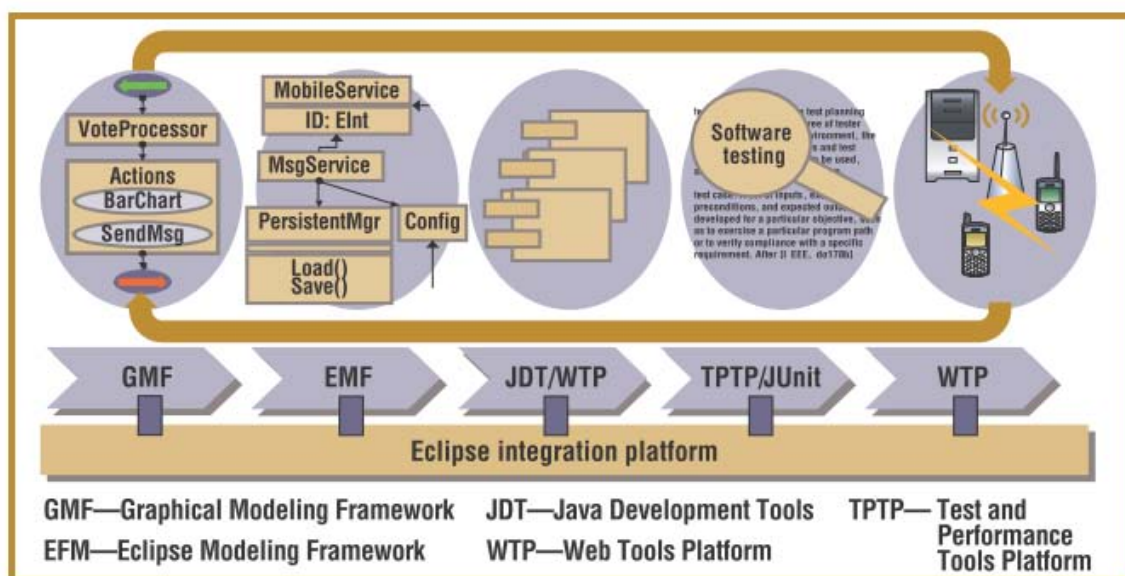
Source: Zhihui Yang and Michael Jiang, Using Eclipse as a Tool-Integration Platform for Software Development, 2007

Eclipse shows a great advantage in their integration ability, which we can conclude from the table above. From the structure aspect, Eclipse has a very important principle which make their great integration power possible, that is” Everything Is a Plug-In.” Eclipse has a basic framework concluding its kernel and the platform consisting of the workbench, workspace, help, and components. It also provides other tools and applications based on this platform. Developers can define their own plug-ins extended from the plug-ins on this platform.

Another importance principle of Eclipse is a specific development

rhythm as follows: 12 months pass between every major Eclipse release, including different stages: “warm-up” (1 month), “milestone builds” (9 months) and “endgame” (1–2 months). This evolution principle make a great contribution on providing a stable and consistent network of plug-ins and innovating it simultaneously.

Zhihui Yang and Michael Jiang (2007) also provided a case of Motorola to show the integration work for life-cycle development. Based on the Eclipse integration platform, there are several frameworks mapping elements for developers add components in it. For example, the Graphic Modeling Framework is the graphic interface, providing developers to program visually; Web Tools Platform plug-ins realizes implementing developers’ own plug-ins for service components as J2EE components and Web services. All the components and services integrated on these frameworks to constitute this system.



Source: Zhihui Yang and Michael Jiang, *Using Eclipse as a Tool-Integration Platform for Software Development*, 2007

2.3. Challenges and requirements of making business sense:

Lots of open source developers concentrated on how to develop out

the software, but not all of them will pay attention to the business sense of open source software, especially considering one of the significant characters of open source that some of developers are motivated by their personal discretion. And even for the companies working on the open source, there still is the same challenge to make a business value. Therefore, basically, related professionals have to make sure of the need before start developing. Dave Thomas (2004) raise up an example to show that some companies don't recognize this requirement. The software of one client had something like 5,000 configuration options and this client even had seven clients. In those 5,000 configurations, there is no doubt of that some of them lacking real world functions.

Moreover, RGF(2007) summarize the requirement of making business value for open source for the enterprises.

Each must meet all enterprise technological requirements (such as architectural compatibility, functionality, and reliability).

Each must offer TCO characteristics that are demonstrably at least competitive with those of comparable proprietary or closed source solutions.

Each must address a specific business need and/or solve a specific business problem;
--

Each must be offered by a vendor and/or vendor partner that also offers commercial, "enterprise-class" support.

Each must demonstrate evidence of long-term market viability, such as community, developer, and/or industry support.
--

3. Conclusions

Software ecosystem consists of business units, the activities of those business units and also the relationships among them. As an new type of ecosystem, Open source ecosystem can be studied for different perspectives, including community, technological/component, and business

values. Open source as a community, it is often started by one person or one group, motivated by assignment or personal discretion. Volunteers join in the group according to their interest and capabilities. Multi projects connected to others to forming the open source community. From the components aspects, framework of eclipse system provided by Zihui Yang and Michael Jiang shows the possibility of realizing an open source ecosystem technologically. To make business value of open source, there are challenges and requirements. Every open source project should have practical need.

4. References

1. *Dave Thomas and Andy Hunt, Open Source Ecosystems, IEEE Computer Society 2004*
2. *David G. Messerschmitt and Clemens Szyperski, Software Ecosystem: Understanding and Indispensable Technology and Industry, 2003.*
3. *Gunnar Stevens and Sebastian Draxler, Appropriation of the Eclipse Ecosystem: Local Integration of Global Network Production, M. Lewkowicz et al. (eds.), Proceedings of COOP 2010, Computer Supported Cooperative Work, DOI10.1007/978-1-84996-211-7_16,*
4. *Jan Bosch, From Software Product Lines to Software Ecosystems, the 13th International Software Product Line Conference (SPLC 2009) August 24-28, 2009, San Francisco, CA*
5. *Jan Bosch¹ and Petra Bosch-Sijtsema, From Integration to Composition: On the Impact of Software Product Lines, Global Development and Ecosystems, Journal of Systems and Software, June 2009.*
6. *Matt Asay, VP, Business Development, Red Hat Summit 2009, Alfresco September 2, 2009*
7. *Michael Weiss, open source ecosystem, <http://opensourceecosystems.com>*
8. *Robert France Group, The Open Source Ecosystem Matures: Observations and Recommendations, Aug 2007*
9. *Slinger Jansen, A Sense of Community: A Research Agenda for Software Ecosystems, 2009*
10. *Walt Scacchi, Joseph Feller, Brian Fitzgerald, Scott Hissam and Karim Lakhani, Guest Editorial Understanding Free/Open Source Software, Development Processes SOFTWARE PROCESS IMPROVEMENT AND PRACTICE, Softw. Process Improve. Pract. 2006; 11: 95–105*
11. *Zihui Yang and Michael Jiang, Using Eclipse as a Tool-Integration Platform for Software Development, IEEE SOFTWARE, 2007*