Service Science Perspective for Open Source Project Survivability

Masashi Ono
Department of Value and Decision Science
Tokyo Institute of Technology
babylon5@valdes.titech.ac.jp

Abstract
With the growing economic importance of Open Source, many researchers have done great efforts for understanding Open Source Software itself and its communities. One of themes which is becoming to get strong attention is “Open Source Software Project Survivability”, which researches the evaluation method of future prospections of open source software communities. Some good works have already been done adopting the concept of ecology. But we have to say that regardless of it persuasive empirical test result using logistic regression analysis data analysis, theoretical appropriateness is not still validated. So, to generalize existing theory, this paper experimentally analyze the true nature of Viability Index using the perspective of Service System. And as its result, it is cleared that Viability Index can almost measure whole OSS community activity effectively.

1. Introduction
Open Source Software (OSS) is now an ordinary existence for all of us who use computer-equipped devices including smart phones or computers themselves, though many of them might not realize the fact. It has a long history, almost same as command line interface, and is explained in Wikipedia, “Open-source software (OSS) is computer software with its source code made available and licensed with a license in which the copyright holder provides the rights to study, change and distribute the software to anyone and for any purpose” and that “Open-source software is very often developed in a public, collaborative manner.” It is proudly called Free/Libre and Open Source Software (FLOSS) compared to corporate-made commercial software. A lot of companies even in non-computer industries are paying strong attentions and investing in their human and capital resources. Typical case is GE, General Electric Company, which made strategic investment of approximately $105 million, representing a 10% equity stake in Pivotal. And this year, the existence of Microsoft surprised many participants of OSCON 2014.

With this increasing company involvement in OSS development and adoption, many researchers have investigated how companies should make use of OSS to gain profits and how to evaluate its communities. Krogh and Hippel [1] are some of great pioneers to investigate OSS from the view of user innovation. Crowston and Howison [2] and others investigated organizational structures of Open Source communities and got clear view regarding “Healthiness” of OSS communities. They are now reviewing the progress of OSS related research activities done by academic researchers using systematic reviewed model named “IMOI (input mediator output input) model” [3], because it has become one of major fields of study and many researchers in various disciplines have done and are doing their works and publishing in many resources.

Based on these research assets, Raja and Tretter [5] succeeded their efforts and has defined and validated multidimensional measure of OSS project survivability named Project Viability. Project Viability has three dimensions: vigor, resilience, and organization. They have defined each of these dimensions and formulated an index named Viability Index (VI) to combine all three dimensions. They have referred to OSS survivability as “the ability of a project to provide functionality and support to its users.” They believed that the measure of project survivability provides a means to evaluate the available OSS projects. So, they have derived the measure under the influence of ecology literature like Costanza and Mageau [6] or Ulanowicz [7] where the survivability of natural ecosystems is measured in terms of their viability.

Figure 1. Three dimensions of Project Viability

Even with the view of other disciplines (e.g. Selectorate theory [8] in political science), their belief...
intuitively seems very persuasive. And their evaluation result using logistic regression analysis makes us difficult to doubt it. It might be possible to say that existing Validity Index satisfies a necessary condition at least.

But as they themselves admit it, they got their evaluation data only from the SourceForge.net data warehouse, which means that it is a kind of statitical analysis and restricted for limited environment use. So, for example, it is difficult for pioneers’ sponsor companies of OpenStack to make efficient use of their theoretical perspective or to evaluate the risk of community explosion caused by its rapid expansion. To generalize their Viability Index theory, thinking of sufficient conditions with the method of Abstraction [22] is useful for the generalization of its theory.

And from the view of reinforcing the theoretical backbone for generalization, it might also be one way to get the opinion from OSS participants. For example, business associates who participated at a community meeting called “Summit” of certain prospective OSS project reported its atmosphere, participants’ age (experience), discussing method and contents and management style, which is important for his mind. Or, the Commit Size Distribution [23] is also useful to understand a certain OSS community’s characteristic.

In these contexts, this paper adopts the holistic system thinking analysis approach (especially, Service Science) and make a revised theoretical model to reinforce Project Viability. Section 2 provides a brief overview of Service Science and its current relation with OSS and OSS Communities. Section 3 defines the revised measure of OSS project survivability. Section 4 discusses its persuasiveness using concrete example of recent OSS related topics. And conclusions are given as Section 5.

2. Service Science and its relation with OSS

There is no clear definition of “Services” and “Service Science”. They are umbrella terms covering diversity kinds of disciplines. This paper uses the definition of Service Science Management and Engineering (SSME) for Service Science perspective [13], because it is designed for disjointing, analyzing and constructing Service Systems and OSS community is expected to be treated as a Service System.

2.1. What is Service Science?

IBM had introduced a term, Service science, management, and engineering (SSME), to describe Service Science more precisely. Today, most of academic researchers use it instead of Service Science when making essential discussions. In Wikipedia, the free encyclopedia on the Internet, SSME is explained as an interdisciplinary approach to the study, design, and implementation of services systems – complex systems in which specific arrangement of people and technologies take actions that provide value for others. For Jim Spohrer, IBM Almaden Research Center, who invented SSME concept, Service Science is the study of value co-creation interactions among entities, known as service systems. Service Science was designed using holistic systems thinking approach among entities, known as service systems. Service Science was designed using holistic systems thinking approach (e.g. “Systematic reviewed model” named “IMOI” model [3]).

Considering this similar perspective, this paper adopts the view of Service-Dominant Logic [10]: the fundamental basis of exchange, that is, service is exchanged for service: all firms are service firms; all markets are centered on the exchange of service, and all economies and societies are service based.

Spohrer et al. [11] briefly explains Service science as: “Service science can be thought of as a mashup or integration of many areas of study known as service management, service marketing, service operations, service engineering, service computing, service human resources management, service economics, management of service innovation, service supply chain and controlling (eSourcing), and others.”

2.2. Service Science and OSS

Spohrer et al. [11] naturally mention Open Source Software communities as: “The normative worldview is that of populations of many types of service systems interacting to co-create value. The types of resources available and the way value is judged differ greatly depending on the type of service systems under study. For example, service systems include businesses, government agencies, people, families, community groups, and open source communities to name just a few. These types of service systems interact (normatively, and certainly not always) to co-create value ranging from monetary value to reputation value, and many other types of value.” So, we can say OSS and OSS communities have already been a part of Service science. The point is how to clearly describe them to achieve our purpose, namely, modifying Project Viability model. It is quite agreeable for this observation but there exists no papers which makes a clear explanation for it.

This paper also has no objection but has only one remark before introducing Service Science perspective for OSS communities. As Maglio and Spohrer themselves mention, value is “usually” defined within economic perspective. It is quite right when we use
Service Science as business tool. But there exists no evidence or proof that OSS community is pure economic activity. The reason why OSS community was mentioned above paper is that the value in OSS community may be little bit different from ordinary economic entities.

So, what is the characteristic of OSS community? By Riehle [12], OSS has changed the rules of the game, significantly impacting the economic behavior of stakeholders in the software ecosystem. In his blog post titled “Open Source Business Research at OWF 2010”, Riehle opened his research result that there exists three key roles that involved OSS community members play.

- Producers
- Users/customers
- Laborers

For example, software developers are part of the procedures as long as it is volunteer work. And about producers, there exist three main types of OSS projects and business-model in real world that expressed with {Who, What, How, Why}

- Who: volunteer or corporate (employed) work
- What: commercial or non-commercial software
- How: community owned or proprietary
- Why: out of altruism or for-profit

The “what” dimension tries to capture whether there is significant revenue potential in the software being produced, which typically implies that commercial parties will join the table. In the “why” dimension, “altruistic” is a catch-all for any motivation that is not strictly for profit, for example, fun or altruism.

Based on these characteristics Riehle categorize OSS communities in three types. IT consultancy company 451 group adopted it.

2.3. Service Science principles for OSS

This time, based on our purpose to reinforce OSS Viability Index for business perspective evaluation, we adopt Maglio’s and Spohrer’s four basic principles [13], which systematically explore value-proposition design as considered in Introduction of this paper. Service Science principles combine organization and human understanding with business and technological understanding to categorize and explain service system, including how the interact and evolve to create value, which goal is to apply scientific approach to advance design and innovation in service systems. Their principle’s foundation is service-dominant logic (SDL). Their core principles center on the way value is computed within and among entities, how interaction is based on access to resources and their capabilities, and on how value computation and interaction depend on symbol processing and language guided by mutually agreed-to value propositions.

Following SDL worldview, Maglio and Spohrer consider an “economic” entity to be a collection of resources, including people, technologies, organizations, and information [14]. SDL identifies two types of resources, namely operant and operand: operant resources, such as people and businesses, operate on operand resources, such as technology (using tools) and information (symbolic processing); thus, to first approximation, their four types of resources – people, technology, organizations, and shared information – are simply kinds of SDL’s two types of resources. And in this paper, the above “economic” is erased to cover whole OSS activities. The main motivation of OSS community members does not directly come from economic success.

### Table 1. Four principles of Service science

<table>
<thead>
<tr>
<th>First principle of service science:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service system entities dynamically configure four types of resources: people, technologies, organizations, and information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Second principle of service science:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service system entities compute value given the concerns of multiple stakeholders.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Third principle of service science:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The access rights associated with customer and provider resources are reconfigured by mutually agreed to value propositions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fourth principle of service science:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service system entities compute and coordinate actions with others through symbolic processes of valuing and symbolic processes of communicating.</td>
</tr>
</tbody>
</table>

In the four basic principles, at its core, a value proposition defines the pattern of shared access to resources among stakeholders over time. The four primary stakeholders are customer, provider, authority, and competitor. As for OSS community analysis, one type of competitors is Closed Source Software (CSS) rather than other OSSs. For example, this paper takes a view that AWS, not CloudStack, mainly brought up OpenStack, which many companies participated in a short time.

2.3. Service Science models for OSS

Based on our purpose and principles, this paper also adopts this time the concept of “Service Chain”, which includes supply chain as a special case, as our
model. It is being advocated by many researchers like Deguchi [15], Jung [16], and many others, which is also a method to construct systems as chains of services.

We use this model because Open Source communities are constructed of many entities; Founders, Core developers, Release coordinators, Codevelopers, Active users, and Passive users [3]. And Governments, companies, and many others affect them. And it is suitable to make dynamic analysis.

Figure 2. Service Chain (Healthcare’s case)

One of the superior features of Service chain model is that we can change our scope, namely, granularity, as we wish. One service system can be constructed with many service systems. Each of them has relationship with others. Service chain can express this fact with clear and dynamic image.

As we have prepared for making our tool, we start constructing our mental model at first. Like many OSS researchers have reported and companies themselves proudly declare their enormous commitments to Open Source communities (e.g. Red Hat for OpenStack), companies and their employees, OSS communities rely much on them now. So, it is very helpful for us to be able to depict each person as minimum element of Service chain model, namely, chain itself.

Figure 3. Basic model of OSS Service Chain

About human entities, boldness of line means the power of influence to others and size of circle means the amount of information which the entity has. For example, the biggest and boldest lined circle in Company A is CEO or like such entity. The biggest and boldest lined circle in Community 1 is the Founder of OSS or like such existence.

But there is one big critical deflection for this model. Its information dissemination method is based on human-to-human interactive communication. Today, CEO of many companies sometimes expresses his or her messages through e-mails and Linus Torvalds, the founder of Linux community, also expresses his plan mainly through e-mails. And based on their behavior, other members also have a chance to express their idea or like such things to many members through e-mails.

So we introduce the concept of “Openness” to our basic model. It defines how widely and openly information is sent. If its Openness is high like Linux community, the organization is clear or white colored. If Openness is low like North Korea, its circle is colored heavily black. In some way, it might be good for us to regard these color shade as a viscous property of fluid. Information can easily disperse widely when a fluid filling some organization is like “air”.

Figure 4. Revised model of OSS Service Chain

But this model is not what we like, because while investigating complex system, human society is not a mere ecological system. It has social or political structure and, even in interactive communications, person’s tendency for dispersing information to others still exists, so it also might be good for us to shade human entity like organizations.

Since “Service Chain” is based on SDL, we can observe above figures from the view of value co-creation. And then from the Riehle’s explanation, namely, “fun or altruism”, two phrases are brought birth.

- “fun” to “technology”
- “altruism” for “organization”
It seems to be just brought here with the connection of words game but in fact, is very persuasive when being observed from people’s true nature. Because economic activity is not at least direct motivation for its members and they are programmers. Yes, they are programmers because they don’t hate to do programming and they have skill sets to do it. How does a person become a programmer? They know the importance and power of technology. They also know the limit of his or her single work and importance to learn from others. Paul Maritz, Pivotal CEO, says that Software is, in its nature, a team sport. So, that OSS community members feel fun to technology and also altruism for organization, namely cooperation and education to keep it exist, is important for them. This is the features that is beyond the scope of the perspective brought by ecology and economics.

3. Revised measure of OSS project

Before defining a revised measure, this paper outlines existing OSS project survivability measure invented by Raja and Tretter [5]. They did so because there didn’t exist appropriate evaluation metric for OSS project survivability. But it is also applicable to CSS project. It evaluates the nature of software project very well.

3.1. Existing OSS project survivability measure

OSS project survivability, Project Viability, derived the measure based on ecology literature, where the survivability of natural ecosystems can be measured in terms their viability. So, they defined Project Viability as the ability of a project to grow and maintain its structure in the presence of perturbations. The three dimensions of project viability are vigor, resilience, and organization. Vigor is the ability of a project to grow, resilience is the ability of a project to recover from disturbances, and organization is the structure exhibited in the project. The three dimensions represent three distinct characteristics of project viability. And for wider application of the measure, they made it independent of technology and the programming environment.

3.1.1. Vigor. They defined vigor as the ability of a project to evolve over a period of time. It is a measure of project growth or throughput. The vigor of a project will change through its life cycle. During the development phase, vigor represents the addition of basic functionality required for successful transition to the next phase of its life cycle. During the maintenance phase, vigor represents incremental changes incorporated to add new functionality to the project. While comparing different projects, growth can only be compared as a function of time. The amount of functionality added can increase with age.

3.1.2. Resilience. Resilience is the ability of a project to respond to internal and external perturbations. These perturbations refer to changes in the operating environment of the project. For software projects, such a disturbance can be artificial or natural. For instance, when Microsoft launches a new operating system, software applications have to adapt to accommodate the new platform. This is a form of disturbance in the operating environment of the project. OSS projects need to make such changes throughout their life cycle to stay relevant. Internal disturbance are changes that emerge from inside the project, for example, a leading team member leaves the group or a module crashes. Resilience refers to the ability of the project to react to the changes in its environment promptly. In a corporate environment, high reaction time can translate into high operational costs. Therefore, resilience is an important component of the project viability, especially for projects in the operational phase of their life cycle.

In the OSS domain, there is no legal or contractual binding that would compel the user to continue using a project. Therefore, low resilience can cause users to switch to other projects with better response time. A decline in the number of end users can affect the donations, popularity, and quality of a project. Poor resilience can also discourage programmers from contributing to a project, thereby causing a reduction in the available development and maintenance effort and eventually affecting the project viability. So, they measure resilience in terms of response time. A shorter response time translate to high resilience.
3.1.3. Organization. They defined Organization as the amount of structure in a project. The organization of a natural system refers to the number and diversity of interactions between the project members and the information exchange among them. A highly organized project is characterized by a high diversity of specialized members and their corresponding specialized interactions. Organization decreases as the diversity of the members and the specialization of the information exchange decreases.

Consider a natural system containing species that feed on only one or two prey and are in turn prey to only one or two other species. This system will have high organization compared to a system with the same number of generalist feeders with multiple pathways to exchange between them [17]. An OSS project with a large group of specialist developers and maintainers to respond to specific problems will have high organization compared to a project with equal number of general-purpose programmers.

Measuring project organization requires knowing both the diversity and magnitude of the interactions within a project. Conte et al. [18] measure the project team efficiency in terms of the communication channels available to them. They point out that the random assignment of tasks and the single assignment of task are both ineffective methods. For optimum utilization of resources, there needs to be core groups of developers to monitor tasks. The measure of organization captures this characteristic of a project.

Information theory offers system level measurement of interactions as a means to measuring system organization [19]. One such measurement is Average Mutual Information (AMI). AMI refers to the amount of information that is available on an element of a system, given the value of another element. It has been adopted in many disciplines, e.g., biology, engineering, and ecology, as a valid measure of organization [19].

3.2. OSS community as Service System

To understand the true nature of someone’s brief, it sometimes be useful to review it using holistic systems thinking perspective. So, as first step, this paper construct OSS community conceptual system blueprint.

OSS community is, like other Service Systems, a real existence under physical law. Then, we can apply concrete systems thinking modeling method to it. It is a kind of system with mechanism of elements and the relations between them. Based on its mechanism, OSS community does internal activities and puts off output.

![OSS community system concept](image)

A kind of Service Systems, OSS community consists of the four resources, namely, people, organization, technology and information, and works based on the four principles. The existence of competitor becomes one kind of input data. As for input data, requests from users also can be one kinds of input data, though it is also possible to regard it as in-system generated data when users are regarded as an element of OSS community system. Output is OSS program itself.

And based on the four principles of Service System, internal activities occur. Examples of activities are commits of Source code, discussions in mailing list, and off-line meeting. So, the research of the commit size distribution [23] is very productive to understand the characteristic of OSS community system.

And if compared to internal combustion engine system, in case of OSS community system, main fuel to drive the system is not only economic motivation. OSS community uses a kinds of hybrid drive engine system with “fun (to technology)”, “altruism (for organization)” and economic motivation. Of course, it also has a face of complex system, so precisely, the modeling with Service Chain System is more suitable to analyze dynamically the specific characters of OSS.

3.3. Viability Index in Service Science

Then, what does existing Viability Index, which consists of Vigor, Resilience and Organization, evaluate?

3.3.1. Vigor. Vigor is defined as the ability of a project to evolve over a period of time. To put it concretely, Vigor is measured with versions released per year. New versions are indicative of functional growth of a project. It means that Vigor measures OSS source code and programs itself.

Namely, Vigor measures a part of Output of the OSS community system.
3.3.2. Resilience. Resilience is the ability of a project to respond to internal and external perturbations. To put it concretely, Resilience is measured with the response time to artifact request. Time taken to complete artifact requests, indicates the ability of the project to handle changes that occur in its operational environment.

Namely, Resilience measures a part of Input and also a part of internal activities of the OSS community system.

3.1.3. Organization. They defined organization as the amount of structure in a project. To put it concretely, Organization is measured with the average mutual information. The structure of the artifact management process reflects the organization of project.

Namely, Organization measures a part of the mechanism of the OSS community system. It is backed by the fact that the only Organization is not directly calculated using time period defined by Raja and Tretter [5].

3.4. Revised Viability Index in Service Science

Existing Viability Index (VI) is defined below as mathematical formula.

\[ VI = a + bV + cR + dO \]  (1)

\( a, b, c, d : \) constant values
\( V: \) Vigor
\( R: \) Resilience
\( O: \) Organization

It is not necessary to destroy this formula. As shown by considering the meaning of Viability Index in the OSS community system shown above, the formula is well balanced to measure the current status of the system. Therefore, the evaluation data of the logistic regression analysis by Raja and Tretter [5] showed good result.

But, to forecast the OSS project survivability for general business purpose and investment, it is more desirable to add more evaluation metrics to it. They are; Competitors to measure a part of Input; Technology to measure a part of the system mechanism of the OSS community system; Information to measure a part of the internal activities; User Support to measure a part of Output. By adding these metrics, the OSS community System becomes able to be evaluated from the whole points of view.

Therefore, new Viability Index (VI) is defined below.

\[ VI = a + d(O+T) + eC + b(V+U) + c(R+I) \]  (2)

\( a, b, c, d, e : \) constant values
\( O: \) Organization
\( T: \) Technology
\( C: \) Competitors
\( V: \) Vigor
\( U: \) User Support
\( R: \) Resilience
\( I: \) Information

Its design concept is like this. OSS community system has its internal mechanism, and gets Input, then performs internal activities and at last, provides Output. So, new VI consists of four dimensions, adding one more dimension to three dimensions of existing VI. As for the OSS projects data analysis extracted from SourceForge.net server, T, e, C, U, I becomes the fixed number because the fact that they exist on SourceForge.net leads to its result. So, VI can still work as effective index indicator by only changing it numerical calculated number.

3.4.1. Technology. Technology is added to measure the possibility of evolution regarding computational assets or OSS itself. Organization is a good metric to check mechanism. But it doesn't work with high performance if there is little energy to drive it. And to measure the amount of energy this paper eyes on evolution. It stimulates the mind of developers, namely, “technology for fun”, and works OSS community system well. Say in different words, if there is no room for evolution, developer loses his or her motivation to participate in OSS development and leaves its community.

3.4.2. Competitors. Competitors is added to measure the motivation to maintain OSS community and provide services (values) including OSS itself to value-co-creators, which are OSS sponsor companies, user companies, users, developers, and so on. Its concept also comes from ecology, namely, the struggle for existence. And competition in this context includes requirements from users because some competitions occur for the purpose of getting users.

3.4.3. User Support. User Support is added to measure non-physical asset provided to OSS users. From the perspective of Service Science, Version-up software is not the only answer to provide value. The last goal of software development is not providing Version-up software, but providing user satisfaction. So, for example, system integration companies can exist.
3.4.4. Information. Information is added to measure core internal activities. Resilience also measures a part of internal activities, but it is strongly focused on non-core activities because it is also designed to measure small part of Input to cover all of OSS community system activities. So, even though Resilience is useful metric to measure internal activities, new VI also adds Information to measure internal activities. But to cover its system status probe with Mutual Exclusive and Collective Exhaustive (MECE) way, Information is focused on measuring core internal activities, say, source code commit number and size of OSS.

4. Discussions

About the fact that OSS community can be processed as one kind of Service Systems, there seems little discussion occurred for it. Almost all people have some kinds of relations with OSS. Services are software and support service for their users. They, user companies, construct computer systems using OSS and provide their service to end users of their computer systems. Almost all organizations and technologists have some kinds of relations. And OSS community basically has the policy to open all information in public. It is easy to image and validate this flow using holistic system thinking view.

As for new Viability Index, it might be helpful to discuss newly revised Viability Index using concrete examples.

But before doing so, it is necessary to explain the reason and purpose of this approach beforehand. Sadly, it is impossible to get concrete real numerical number for revised Viability index because unlike existing Viability Index, the targets of new Viability Index is not able to use common server or something. The constant values of existing Viability Index is derived from a mass of statistical data of many OSS communities. So, adding consideration that OSS is essentially dynamic entities, it might be good to search other method. (e.g. Dynamic modeling, Agent based simulation, and etc.) And it is good for the existing Viability Index user to pay attention to the situation of SourceForge.net because like Zenoss, some promising OSS projects move their activities to their own sites and use SourceForge.net only for marketing promotion method.

4.1. Technology

One example is text editor software. When Windows, Linux, and other operating systems started providing first commercial versions, text editor software has a lot of things to improve. GUI (Graphical User Interface) was a typical area.

Many OSS projects occurred and provided a lot of good software. But today, most computer users don’t install text editor software any more. Instead they use built-in text editor software provided by Windows, Linux, and other OSs and many OSS projects for text editor software disappeared. Sometimes, someone starts establishing new project for his or her special needs but also disappears in less than several years suddenly. To adapt to these kinds of OSS projects, it is good to use Technology metric to eliminate such existence.

Another example is the Cloud Foundry founded by IBM, EMC, VMware, HP, Rackspace, SAP and Pivotal in February, 2014. It has expanded to 33 sponsors and 42 contributing companies in July, 2014. It is growing so fast that existing Organization metric doesn’t show good value. This rapid expansion has occurred for the bright prospect for future growth. In such case, it is good to introduce Technology metric.

4.2. Competitors

In ecology, the struggle of existence is important. In OSS world, proper competition is good to promote product development. For example, Zenoss Core competes with Nagios, Groundwork, Zabbix, and so on. OpenStack came into existence from the dissatisfaction to CloudStack as some kinds of forking but is rather growing to compete with Amazon AWS.

But too much competition is not good for OSS project survivability. In case of OpenStack, severe competition with Amazon AWS has caused rapid expansion and IT research companies like IDC and Gartner warns that it is not good for its community’s health. Pivotal, a founder of the Cloud Foundry Foundation, also faced concern for this phenomenon and realized that member’s average age of OpenStack is much younger than those of Linux Foundation. It is not opened in detail but in some newspaper’s interview articles, Pivotal adopts, in a way, apprenticeship, which seasoned employee mentors younger programmer.

Another example is Ceph open source software, which was being developed by InkTank, Inc. whose CTO was the inventor of Ceph, and is adopted by several Linux distributions including RHEL, Debian, Ubuntu, SUSE, and so on. After competing with Red Hat’s Gluster FS software for several years, InkTank was acquired by Red Hat on May 2014. The Attachmate Group, who is the owner of SuSE, might now in difficult position.

So, Competitors metric affects the OSS project survivability.
4.3. User Support

In terms of Service Science, OSS is not the only material that provides value, in this case, customer satisfaction. User Support is also important. Zenoss is very famous for its slow development of next version. For example, they help community users to use workaround measure instead of providing new improved version. Zenoss now owns its original web site. For such entity, it is useful to introduce User Support metric adding to existing Vigor metric.

By the way, some OSS has possibility to stop providing new version suddenly, if, for example, most core developers belong to a company and it fails its OSS business. In that case, users need to continue using existing version. Small sized software developed by a single developer sometimes disappears for his or her private reason.

4.4. Information

Resiliency is good metric to measure internal activities of OSS community but it is designed to monitor the ability of a project to respond to internal and external perturbations. So, it is appropriate to monitor the performance of development team. As already mentioned, OSS projects sometimes have to fight for the survival for their existence. So development efficiency becomes also important in OSS project survivability. Monitoring the commit number and size distribution as Information is important. And it also contributes to effective management of development team.

5. Conclusions

This paper has made it clear that OSS community can be regarded one type of Service Systems and that the idea of the four resources and the four principles of Service Systems is useful to analyze the nature of OSS community. “fun to technology” and “altruism for organization” are characteristic features of OSS community members. Modified multidimensional measure of OSS project survivability has been defined and also understood that existing measure was well organized from holistic systems thinking view but has a room for more general use improvement. Modified Project Viability is defined in terms of vigor, resilience, organization, competitors, technology, information, and user support. But this paper could not find the way to do quantitative empirical validation so improved research or invention of another evaluation method is strongly desired.

6. References


