RC24723 (C0901-005) January 13, 2009
Computer Science

IBM Research Report

Service Quality Evaluation Method for Community-based Software Outsourcing Process

Huimin Jiang[1], Alice Liu[2], Zhongjie Wang[1], Shu Liu[1]

[1]School of Computer Science and Technology
Harbin Institute of Technology
Heilongjiang
Harbin 150001
P.R. China

[2]China Research Laboratory
Building 19, Zhouguancun Software Park
8 Dongbeiwang West Road, Haidian District
Beijing, 100094
P.R.China

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Service Quality Evaluation Method for Community-based Software Outsourcing Process

Huimin Jiang[1], Alice Liu[2], Zhongjie Wang[1], Shu Liu[1]

(1 School of Computer Science and Technology, Harbin Institute of Technology, Heilongjiang, Harbin, China 150001)

(2 IBM China Research lab, Beijing, China, 100094)

Abstract: Outsourcing software development to the community developers is a promising model to help reduce software development cost and improve software quality. In this paper, we present a method to evaluate the quality of service in the managing community-based software outsourcing process. There are three types of objects whose service quality need to be evaluated, i.e. service behaviors, service products, and service providers. For each type of object, there are five dimensions of quality indicators, i.e. time and efficiency, price and cost, quality of service content, resources and conditions, reputation and risk. Based on these dimensions, we built a set of quality indicators and the corresponding measurement methods. In our work, we adopted the traditional AHP method to calculate the total quality of each type of object. Call-For-Implementation is community-based software development method put forward by IBM china research lab. In this paper, we take Call-For-Implementation as an example to introduce how to apply our proposed service evaluation method. A prototype is developed to support the evaluation process and exhibit results of quality evaluation. The main contribution of this paper is that an objective and dynamic service quality computation method is proposed to help evaluate the quality of outsourcing software management service.

Key words: community-based software development Call-For-Implementation (CFI) quality indicators service quality evaluation

1 Introduction

With the rapid socio-economic development, service becomes more and more popular. At the same time, service quality is increasingly attracting wide attentions. The reason why service quality becomes so important is that the higher value of service quality can lead to higher customer satisfaction, and ultimately result in higher revenue growth and profitability.

More and more enterprises are looking to outsource their software development [1, 2] to other companies, respond to the pressures such as development costs, human resources access, new market development, or building business competencies. Recently, to outsource software development to the community developers through web has been recognized by more and more people [8]. We called this outsourcing approach as community-based software model in this paper. Call-For-Implementation is a community-based software development method put forward by IBM China Research Lab [8]. CFI method proposes that software owners can leverage the community resources to develop software. In the procedure of software development, some labor intensive works, such as coding and testing tasks can be outsourced to the community developers, such as college students, high school students, and programming freelancers. By leveraging a large number of community developers, CFI method can help dramatically reduce software development cost. However, the quality of community developers and project manager (PM) will greatly impact the quality of software delivered by using CFI method. It is necessary to provide a good mechanism to evaluate the service quality from multiple dimensions for CFI project management process. For example, the software owner needs mechanisms to discover suitable PM, and PM needs to discover suitable programmers and testers (P/T). To accomplish this, CFI platform should like the general service to have the service evaluate method to offer sufficient quality information for the improvement of the efficiency and effectiveness of the community-based outsourcing service process [3] [4].

In the past, the evaluation of service quality in the traditional service is done after the service, and the data for calculation is from the customer's perspective. In this paper, the evaluation is objective and done in real-time. The community-based software development will be split by the act of service and then be evaluated separately. By the real-time evaluation of service quality, some problems can be found at an early time, so as to reduce the cost. The objective evaluation makes the result of service quality be more reliable. The separate service quality evaluation of each service behavior split by CFI provides an important reference for the choice of P/T and PM in future community-based software outsourcing service.

Firstly, we should determine the evaluation objects in community-based software outsourcing process. Secondly, we put forward the quality indicators of each object in five dimensions: time and efficiency of service, price and costs of service, quality of service content, resources of service and credit and risk of service. Thirdly, we need to give measurement methods for each quality indicator, adopt traditional AHP and the weighted sum method to calculate the value of service quality of the five dimensions and evaluation objects. At last, the results of quality evaluation would be displayed in a prototype system.

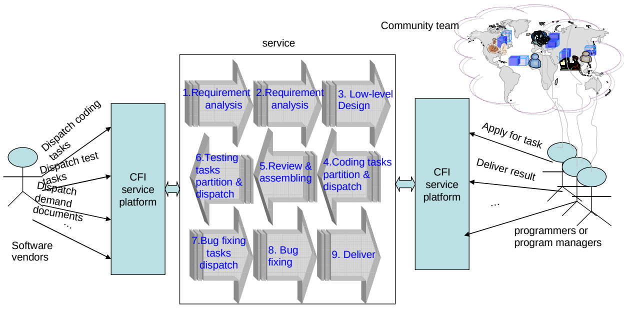
The rest of this paper is organized as follows: Section 2 focuses on the introduction of CFI service platform, which is a case of the community-based software outsourcing service, including its architecture and service process; Section 3 presents the service quality evaluation method in CFI, including the evaluation objects, service quality indicators system and service quality calculation method; Section 4 describes how to display the results of service quality to CFI users. Section 5 is a summary.

2 CFI Service Platform

2.1 The Architecture of CFI

The service CFI provides is a community-based software outsourcing, which is different with traditional outsourcing. In CFI, software owners can be responsible for the needs of front-end analysis and design. CFI also provides methods and tools to help software outsourcing tasks, will be done at different levels, at different stages and different size of the split, which will be split after the tasks can be posted on the Internet. Then the developers can be contracted through the network to choose one or more of the tasks. Fig.1 depicts the overview of the CFI method.

Fig.1 Call-For-Implementation method overview

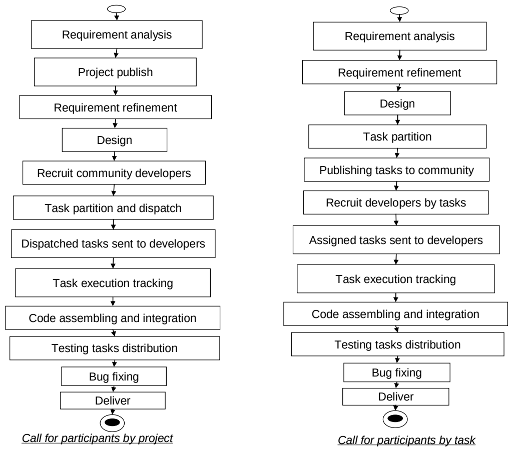


Here, software owners and developers are all the customers for CFI service platform. But in the community-based software outsourcing, software owners are the customers, and software developers, including programmers, testers and program managers, are the service providers.

2.2 The Process of CFI

A community-based software outsourcings project that is developed using the CFI method has to follow the process defined by the CFI method. This process is called the CFI process for short. According to the different community developer's recruiting models, the CFI process is classified into two different types: one is called call for participants by project and the other is called call for participants by task. The first type means that the project team is built before assigning tasks; and the second type means that the community developer is recruited after tasks are partitioned. Fig.2 depicts these two CFI process modes. In this paper, focuses on the latter one.

Fig.2 Two modes of CFI process


Requirement analysis
Project publish
Requirement refinement
Design
Recruit community developers
Task partition and dispatch
Dispatched tasks sent to developers
Task execution tracking
Code assembling and integration
Testing tasks distribution
Bug fixing
Deliver
Call for participants by project
Requirement analysis
Requirement refinement
Design
Task partition
Publishing tasks to community
Recruit developers by tasks
Assigned tasks sent to developers
Task execution tracking
Code assembling and integration
Testing tasks distribution
Bug fixing
Deliver
Call for participants by task

3 Service Quality Evaluation Method

In the community-based software outsourcing service provided by CFI platform, software owner is the customer. As a result, the evaluation method of service quality in outsourcing should follow on the criteria that to measure whether the demands of software owners are violated, if true, what the pitch of violation. The smaller pitch of violation, the better service quality is. On the other hand, the worse service quality is. In this paper, we utilize the satisfaction of software owner-based evaluation method.

3.1 Evaluation Object

In this paper, there are three categories of the evaluation object: service product, service provider and service behavior. Service product refers to the final system/software submitted to the software owner. Not all the services have product, such as logistic export has no product. The service quality of community-based software outsourcing mainly depends on the service quality of finally submitted system/software. Therefore the evaluation of system/software is necessary. Of course, it also has something to do with the quality of submission time, risk and credit. Service provider means the provider of community-based software outsourcing service. However, it makes no sense to purely evaluate the people in software outsourcing service. The service quality value of people in software outsourcing service equals to the quality value of the behavior which this person participated in. So we also have to evaluate the service behavior split by CFI method. Service behavior is a component of software outsourcing, which is independent, centered on the act and reusable. The separate service quality evaluation of each behavior also provides an important reference for the choice of P/T and PM in next time.

Table 1 Evaluation objects in community-based software outsourcing

service behavior	behavior of programmer (C_1)
	behavior of tester (C_2)
	behavior of test use case designers (C_3)
	behavior of PM (C_4)
service provider	programmers
	testers
	test use case designers
	PM
service product	System/software

3.2 Service Quality Indicator System

Here, the service quality indicators system is a three-tier system. The first layer is the service quality (SQ) of the evaluation object. On the second level, there are five dimensions: time and efficiency (Q1), price and costs (Q2), the quality of content (Q3), resources and conditions (Q4), reputation and risk (Q5). The third layer puts forward the specific quality indicators for each dimension. Each layer has a one-to-many mapping relationship with its next layer.

Time and efficiency: measure the period between the start of service and the completion of service, whether the developers timely submit their products, the service providers' responsiveness in the service process and so on.

Price and costs: whether the price or the cost of realization is reasonable.

The quality of content: measure the satisfaction of customer with the content of service, which includes ease of establishment of system/software, access to and use of system/software, customers' personalized experience with this system/software, service attitude, and so on.

Resources and conditions: measure the quality of the resources that supports the service, the quantity to meet and continue to meet, and so on.

Reputation and risk: the quality of reputation of a service is a measurement of its trust worthiness [3], such as the credit of promise of delivery time and quality.

As C_1, C_2 and C_3 have same quality indicators on Q_1, Q_4 and Q_5, so we just show the indicators of C_1.

Table 2 Indicators of C_1

Q_1	q_{11} punctuality	Programmers submit the results of the task on time
	q_{12} development cycle	From start to completion of the development of a task
	q_{13} attendance	There are logs in CFI, require developers to report the progress of the development and problems on time every day
	q_{14} progress control	Whether the developer have applied for an extension, how long he/she applied for, whether the application is accepted
Q_3	q_{31} Rework	How many times asked to rework
	q_{32} attitude	The attitude in the cooperation with PM and other developers
	q_{33} normative code	Whether the code is written in accordance with the requirements of the format
	q_{34} potential error	The potential errors in the code, such as memory leaks, etc
	q_{35} serious error	Include the serious errors have been checked out and not
	q_{36} correctness of code	Include the errors have been checked out and not, except the normative and potential errors
	q_{37} task	Compare with the demands of

	completion	the task to measure the completion of the task
	q_{38} bug fix	Whether all the bugs that have been checked out be fixed
Q_4	q_{41} education	
	q_{42} technical	How many technologies developer mastered
	q_{43} project experience	The developer's project experience
Q5	q_{51} credibility of the quality of service	Measure the honor of developer's commitment to the quality of the task result
	q_{52} credibility of time	Measure the reliability of developer's commitment to the submit time

Table 3 Indicators of C_4

Q_1	q_{11} punctuality	PM submits the results of the project on time
	q_{12} develop cycle	From start to the completion of a project
Q3	q_{31} attitude	The attitude in the cooperation with software owner
	q_{32} coordination ability	a)the relationship with subordinates b)the relationship with customers
	q_{33} quality of the project	As the final submission for software owner is a system/software, so this part refers to the quality of the system/software
Q4	q_{41} education	
	q_{42} project experience	The PM's project experience
Q5	q_{51} credibility of the quality of service	Measure the honor of PM's commitment to the quality of the project
	q_{52} credibility of time	Measure the honor of PM's commitment to the submit time
	q_{53} credibility of cost	Measure the honor of PM's commitment to the budget

Table 4 Indicators of system/software

Q_3	q_{31} the completion of the functional demands	Compare with the demands of the project to measure the completion of the project
	q_{32} The number of unresolved bug	
	q_{33} The number of unresolved serious bug	
	q_{34} ease of use	Include the ease of establishment, access to and use of the system/software, such as the steps needed to use it, and the prompts provided for how to use it
	q_{35} friendly	Customer's personalized experience, the feeling of tailor-made

3.3 Service Quality Calculation Method

In our work, we take CFI as an example to explain how to apply our evaluation method to measure the service quality of community-based software development. So the data for calculating the value of service quality is mainly collected from CFI. There are three ways to collect data: the logs in CFI platform, acquisition by agents, and manual evaluation. Most data can be collected from the logs. Some data do not exist in the log but in CFI platform, these can be collected by agents. Others can be collected as the feedback from people.

In the calculation of service quality, we adopt traditional AHP [5, 6] method with weighted sum [6, 7]. There are mainly three types of ultimate value of service quality (with notation SQ): the SQ of P/T, the SQ of PM and the SQ of outsourcing service. The SQ of developers, testers and PM equals to the quality of their behaviors in certain outsourcing service. The SQ of outsourcing service equals to the weight sum of the SQ of system/software and the value of Q_1 and Q_5 of PM.

Firstly, we should distribute the weight (with notation W_i) for the five dimensions, Q_1, Q_2, Q_3, Q_4 and Q_5 by AHP evaluation method, then distribute the weight(with notation w_{ij}) for the next dimension, q_{ij}.

$$\sum_{i=1}^{5} W_i = 1, \quad \sum_{j=1}^{n} w_{ij} = W_i$$

Secondly, we put forward the calculation method for each quality indicators, q_{ij}. Different indicators have different algorithms. In this paper, the method of calculating the points is that the highest score should be less than or equal to 1 point, and the lowest score have no limit, maybe -1 or even lower. The quality value calculation of C_4 is showed to explain how to calculate the value of quality.

Q_1 time and efficiency:

Constraints: the project interval is [a, c], a: the start time of project; c: the final time of project; the cycle equals to (c-a).

The actual situation: submit at f, the actual development cycle is (f-a), a, c and f all denote the date.

q_{11} punctuality:

if f<=c:

$(c - f)/(c-a)<1/100 \mid q_{11} = 0.9$

$(c - f)/(c-a) >=1/100 \mid q_{11} = 1;$

else:

$(f-c)/(c-a)<5/100 \mid q_{11} = -0.5$

$(f-c)/(c-a)<10/100 \mid q_{11} = -0.6$

(f-c)/(c-a)<15/100 | q_{11} = -0.7

(f-c)/(c-a)<25/100 | q_{11} = -0.9

(f-c)/(c-a)>=25/10 | q_{11} = -1.

q_{12} develop cycle:

$$if\ \ T \ge T_1\ \ q_{12} = 0.8 + \frac{c-f}{c-a},\ \ else\ \ q_{12} = \frac{c-f}{c-a}$$

Q_3 the quality of content:

q_{31} attitude:

The attitude of PM is evaluated subjectively by software owner. If quite good, q_{31}=1; if good, q_{31}=0.8; if general, q_{31}= 0.6; if not good q_{31}= -0.2; if quite bad q_{31}=-0.6.

q_{32} coordination ability:

N_1: the number of complaints from P/T;

N_2: the number of complaints from software owner;

Certainly, the customer's complaints are more important than P/T's.

$$q_{32} = 1 - \frac{N_1}{4} - \frac{3N_2}{4}$$

q_{33} quality of the project: This is a comprehensive quality indicator. This quality value equals to the quality of the system/software finally submitted, which totally depends on the Q_3 value of system/software.

$$Q_3 = \sum_{i=1}^{4} w_{33i} * q_{33i}$$

The w_{3i} and q_{3i} all belongs to the Q_3 of system/software in this community-based software outsourcing service.

A: on behalf of the overall difficulty of the project.

q_{331} the completion of the functional demands：

N: the number of function demanded in requirement document;

N_3: the number of actual completion of the functions demanded.

$$q_{331} = \frac{N_3}{N}$$

q_{332} the number of unresolved bug:

Here, the unresolved bugs refer to the bugs which have been tested but still unresolved after rework (with notation R), as well as the return bugs (with notation H). The return bug means the bug that had not been checked out by testers. Suppose there are n tasks in this outsourcing process. The total number of R equals to the sum of R_j and the total number of H equals to the sum of H_j. (n>j>0) (n: the total number of tasks)

A_j: on the behalf of difficulty of the corresponding task.

$$q_{332} = -\sum_{j=1}^{n} \frac{1}{A_j} H_j R_j$$

q_{333} ease of use：Is evaluated by P/T, if quite good, q_{333}=1; if good, q_{333}=0.8; if general, q_{333}= 0.6; if not good q_{333}= -0.2; if quite bad q_{333}= -0.6.

q_{334} friendly：Is evaluated by P/T, if quite good, q_{334}=1; if good, q_{334}=0.8; if general, q_{334}= 0.6; if not good q_{334}= -0.2; if quite bad q_{334}=-0.6.

Q_4 resources and conditions:

q_{41} education：Refer to the registration information.

If high school education or less, q_{41}=0.2; if graduate, q_{41}=0.6; if master, q_{41}=0.8; if doctor or better, q_{41}=1.

q_{42} project experience:

i: is corresponding to the serial number of projects

t_i: There are three type projects, if is 863, 973, or the same level, t_i =0.4; if the project is a big one t_i =0.2; else t_i =0.05.

k_i：If the PM's role of the corresponding project is PM, k_i=1; if is not PM but one of the team leaders, k_i=0.8; if is one of the most important team members, k_i=0.4; if is just a normal member, k_i= 0.1.

Ni: the number of project belongs to the corresponding type.

$$q_{42} = \sum_{i=1}^{n} N_i t_i k_i;\ \ if\ \ q_{42} \ge 1\ \ q_{42} = 1;\ \ else\ \ q_{42} = q_{42}$$

Q_5 reputation and risk:

q_{51} credibility of the quality of service:

Here should refer to the value of q_{31}, q_{32}, q_{33}, q_{34} in table 4.

$$q_{51} = \frac{4}{10} q_{31} + \frac{4}{10} q_{32} + \frac{1}{10} q_{33} + \frac{1}{10} q_{34}$$

q_{52} credibility of time: depends on the value of q_{11} in table 3.

If q_{11}>0 q_{52}=1; else q_{52}=q_{11}

q_{53} credibility of cost：RMB x Yuan: the budget for the project; RMB y Yuan: the actual cost.

$$if\ \ y \le x\ \ q_{53} = 1,\ \ \ else\ \ q_{53} = \frac{x-y}{x} * 10$$

Thirdly, use equation 1 to calculate the value of Q_i.

$$Q_i = q_{i1} w_{i1} + K + q_{in} w_{in} \qquad \text{Equation 1}$$

(n: the number of indicators under Q_i)

Finally, use equation 2 to calculate the value of SQ for every evaluation object.

$$SQ = Q_1 W_1 + Q_2 W_2 + Q_3 W_3 + Q_4 W_4 + Q_5 W_5 \qquad \text{Equation 2}$$

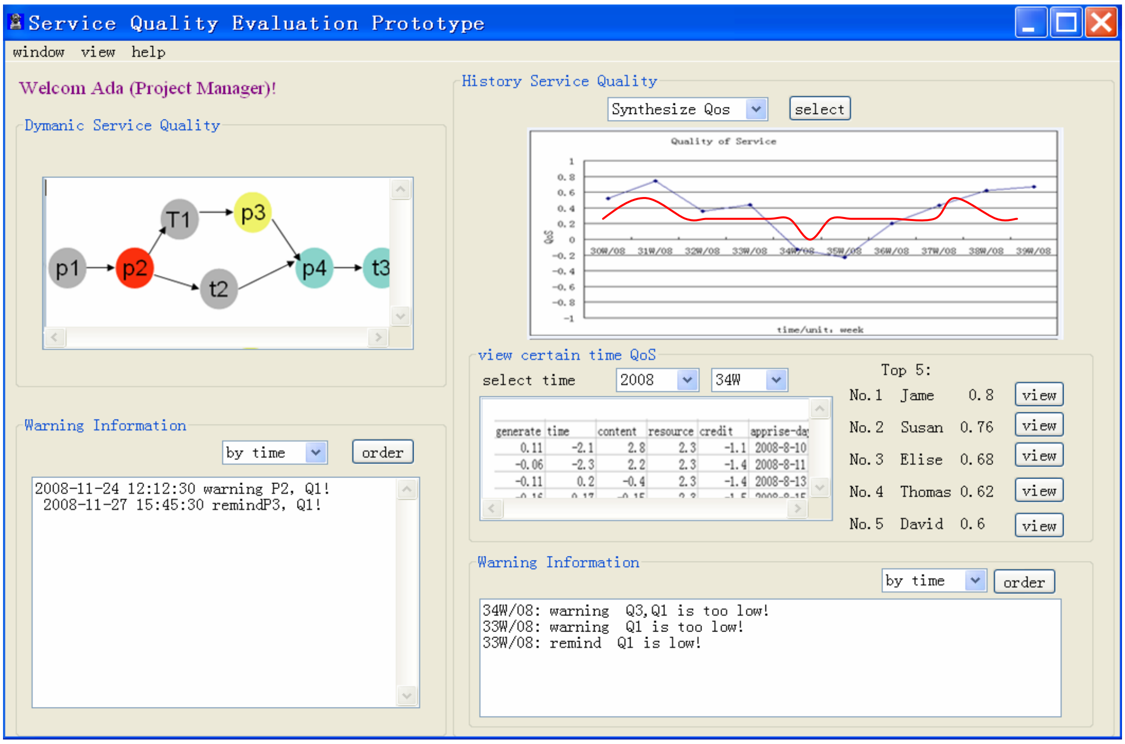
4 Service Quality Evaluation Prototype

This prototype is developed to support the evaluation process and exhibit the result of quality evaluation. There are two kinds of quality value to be displayed, the dynamic value of service quality and history value of service quality. Fig.3 shows what the PM can see after his login. On the left side of the window shows the dynamic service quality of the project which this PM is participating in, including the service behaviors of this outsourcing service and the corresponding warning information. In the "Dynamic service quality" part, there are the service behaviors of this service. The number of behaviors equals to the total number of tasks divided by CFI tools. Logo P is the representative of the behavior software developers participate in. And logo T is the representative of the behavior testers participate in. The number marked in the bottom corner of P and T is corresponding with the serial number of the task. The logo in gray means the finished behavior, blue means unfinished behaviors and red represents the finished behavior with warning, yellow means the unfinished behavior which is threatened. We exhibit the corresponding warning message on the bottom of the window. You can range the message by time or task serial number. If ranged by time, the latest one would be shown in the first.

On the right side of the window shows the history service quality value of this PM. You can choose to view the value of synthesize or time and efficiency, price and costs, system/software, resources and conditions and reputation and risk. The red line is the average service quality value of the similar. If the service quality of this PM is lower than the red line, he would receive some warning. This prototype also provides early warning function according to the history service quality. As we can estimate the future service quality of this object by his history service quality. If the estimated value is lower than the red line, system will give him an early warning. In the "view certain time service quality" part, user can select a certain week to see a specific quality value during this week. On the right of it, shows the top 5 objects whose value is in the first 5 under the selected dimension. Click the button "view", the left small window would show this object's service quality under the selected dimension.

In our research, we not only measure the value of service quality, but also give service quality information to PM to choose P/T, to software owners to choose PM. In the menu, software owners can look for service quality information of any developer or PM by clicking the button "view".

Fig.3 Service Quality Evaluation Prototype (PM login)



Welcom Ada (Project Manager)!

```
2008-11-24 12:12:30 warning P2, Q1!
 2008-11-27 15:45:30 remindP3, Q1!
```

select time 2008 34W

generate	time	content	resource	credit	apprise-day
0.11	-2.1	2.8	2.3	-1.1	2008-8-10
-0.06	-2.3	2.2	2.3	-1.4	2008-8-11
-0.11	0.2	-0.4	2.3	-1.4	2008-8-13

Top 5:

No.1	Jame	0.8
No.2	Susan	0.76
No.3	Elise	0.68
No.4	Thomas	0.62
No.5	David	0.6

```
34W/08: warning  Q3,Q1 is too low!
33W/08: warning  Q1 is too low!
33W/08: remind  Q1 is low!
```

5 Conclusion

In this paper, we present a service quality evaluation method for community-based software development, and take CFI as an example to explain how to apply our proposed service evaluation method. We not only express the satisfaction of software owners with the software/system, but also help P/T and PM monitor their dynamic service quality, which can help them detect problems at earlier stage, remedy the problems timely and greatly reduce cost. The data for evaluation is objective and collected in real-time, because most of the data are from the CFI platform without personal perspective. To a certain extent, this ensures the result is more objective and reliable. At the same time, we also recommend the objects with good service quality to others for their choice and learning. For future work, we will put forward the factors affect the quality of indicators. By doing this, we can help service providers greatly improve their service quality.

Acknowledgement Research works in this paper are supported by the National Natural Science Foundation (NSF) of China (No. 60803091, 60673025) and IBM-HIT Joint Lab 2008-Year Joint Research Project (No.JSP200807004-1).

References

[1] Lacity, M.C. and Wilkocks, L.P. Global Information Technology Outsourcing. Wiley, 2001.

[2] Heeks, R.B., Krishna, S., Nicholson, B., and Sahay, S. Synching or sinking: Global software outsourcing relationships. IEEE Software 18, 2 (Mar. /Apr. 2001), 54–61.

[3]Yutu Liu, Anne H.H.Ngu and Liangzhao Zeng. QoS Computation and Policing in Dynamic Web Service Selection. Thirteenth International World Wide Web Conference Proceedings. p 798-805, 2004.

[4] J.Fan and S.Kambhampati. A snapshot of public Web services. SIGMOD Record, 34(1):24-32,2005

[5] http://www.fync.edu.cn:8080/2006jxms/nzx/msword/nl05.doc.

[6] Vargas LG. An overview of the analytic hierarchy process and its applications[J]. European Journal of Operational Research, 1990, 48(1):2~8.

[7] Yunyan Wu, Zhongsheng Hua, Yong Cha. Calculation of the Weights and the Amalgamation of the Matrixes in AHP Group Decision. Operations Research and Management Science.2003/04/12: 16~22.

[8] Liu Ying, Feng Chenhua, Zhao Wei, Su Hui, Liu Hehui, A Case Study on Community-enabled SOA Application Development, IEEE International Conference on Service-Oriented Computing and Application, June 19-20, 2007, Newport Beach, California.

[9] http://www.topcoder.com