A Quality Evaluation Framework Based on Distribution Measurement in Service Computing Environment

Zhenyu Liu
Shanghai Key Laboratory of Computer Software Evaluating and Testing
Shanghai, China
e-mail: lzy@ssc.stn.sh.cn

***Abstract*— The importance of software quality evaluation has being gradually more and more focused. This paper studies service-based software quality evaluation in service computing environment. As is known to all, the existing evaluation technology adopts concentrated strategy during run-time. The paper compares the difference between the offline and online strategy for distributed system, and analyzing the features in service computing environment furthermore. The paper puts forward a novel quality evaluation measurement model. The attribute factoring technology and corresponding data collection strategy are described. A new evaluation method of distributed quality data acquisition is put forward in the paper, which is based on software testing technology. So, during service runtime operation, the obtained results by test method can be accurate and credible. Finally, this paper puts forward quality framework and related steps which is evaluated by distributed in service computing environment. Through example of manufacturing industry, it shows that the quality evaluation in distribution measurement framework is a effective and trustworthy.**

Keywords-Quality Evaluation; Software Testing; Distributed Measurement; Service Computing

I. INTRODUCTION

With the development of network, more and more software based on distributed architecture, such as Web Service, Service-oriented, Grid Computing SOA and cloud computing. The rapid growth of computer hardware and network, distributed system architecture make software more complicated, especially service-based computing convert the traditional computer and centralized storage approach into distributed architecture according to the demand of end user.

Service-based computing is service computing, which accomplished combining distributed processing, parallel processing and grid computing. The further research is realization of the commercial on view of user's point. Generally, basic principles of service computing are considered using the distribution by many computing resources, rather than the local computer or remote server.

This allows companies to adopt the appropriate resources of applications to access computing and storage resource, according user-demand. [1] As for service, life-cycle consisting of service development, service registration and service delivery. The three stages divided into offline testing and online testing according to the service releases correspond to the quality of service testing [2]. Offline test means the software testing without real environment. The test is executed in development environment. Before validation, this stage mainly verifies service function on the unit testing, system testing and other testing techniques related to code and service interface. Online testing is the design for the service lifetime, which focuses on non-function quality related with testing activities during execution of service.

In service environment, services register and publish using uniform computing platform. These services publish on the distributed environment. The function of service is same in different deploy environment. However, the actual quality of service should be considered when the service is being carried out. Service should be published and deployed in the middleware server and registered in specific server before use. Therefore, quality of service should be evaluated delay to the runtime phase. And continuous test is key approach to acquire original quality data.

In Figure 1, we compared different software phase in various distributed environments. The typical software phase is software development, software testing and system runtime. And the usual environment consists of network-based environment, grid environment and cloud computing environment. From the comparison, some distributed system required to postpone the online tests into run-time. The cloud environment is kind of these distributed systems. The results of software testing are obtained through the evaluation of the quality of service during the online testing.

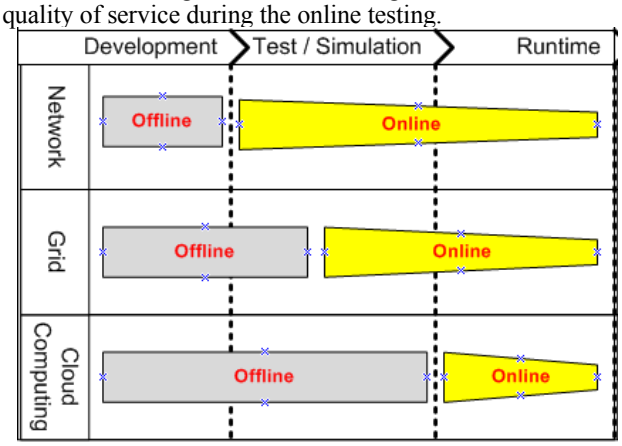

Figure 1. Comparsion of phase in distrubted environment

 ISBN: 978-1-61208-157-1

With the development of software techniques, there is change from the stand-alone application to network-based applications gradually. However, service-based computing software changes current operation and computer method. Therefore software quality is more and more to pay sufficient attention [3]. Also the users of the software quality are constantly changing from the initial needs that meet the functional requirements to non-function quality, such as run-time efficient and convenient.

As the service computing environment is the loose network, the test during the phase of running service in scenario. The scenario is closer to the genuine quality of the software runtime environment. Therefore, there is inconsistency and uncertainty for online quality of service in contrast to development period. Table 1 gives the different development period under the various architectures. In contrast to other architecture comparison, it is evidence that cloud computing environment own run loose-couple management, which mean the online quality is not analyzed and evaluated in stage of development period effectively.

TABLE I. COMPARISION WITH DIFFERENT ARCHITECTURE

Architecture	**Environment and related Stage**		
	Development	***Testing***	***Runtime***
Stand-alone	Stand-alone	Stand-alone	Stand-alone
Browser / Server	Single/Multi	Single/Multi	Network
Application / Midderleware	Single/Multi	Multi	Integrated Management
Cloud Computing	Stand-alone or Single/Multi	Single/Multi	Loose-copuled

Through the above analysis, it is necessary to transfer the validation and testing of service from the testing stage delay to the runtime stage. For the loose network applications, various users on different network nodes access the same services. But the results obtained varied greatly, which is different from the traditional stand-alone architecture. From the results of a single user to determine the software quality is not sufficient. It is also not enough to determine the quality in the same network. It is necessary that the distribution of methods used to measure this evaluation [4].

The main structure of this paper is as follows: The evaluation of quality decompose model in section 2. Section 3 introduces data collection method through testing method. The next part presents the distributed online quality evaluation framework. Finally, there are conclusions and future works.

II. EVALAUTION MODEL

Software quality evaluation model is the approach that evaluate and quality of the quantitative evaluation according to software-related needs. Boehm et al proposed a hierarchical model of software measurement [5]. McCall proposed three-level model to measure, which divided software quality into elements, standards [6]. At present, the evaluation model is main method of measure software features. ISO/IEC TR9126 standard gives a typical measurement model for software product. The proposed model consists of an external measurement metrics, internal measurement metrics and user measurement metric [7].

A. Generic Model

The quality requirements of the software have functional requirements and non-functional requirements. As for quality requirements of system design, the traditional software quality evaluation model is mainly based on quality index in evaluation of quality.

For different scenarios, quality requirement are different according to request user. Especially for cloud computing environments, quality demand of different user is also changing.

However, the existing evaluation model gives the design requirements for the proposed related measurement methods. And measurement methods and their model include some measurement items. Under many circumstance, the data from items adopt the offline process, that is to say, system is not still provided service in actual environment.

Therefore, the quality of evaluation process also meets the unique context during evaluation. A single context is not applicable for various scenarios which involve in the application of measurement models. In practice, the different scenarios will eventually lead to different measurement results.

The typical quality model is proposed by ISO/IEC TR9126 international standards. This standard consists of four parts: a generic measurement model and three measurement models from a different perspective, which is internal metrics and external metrics, and user metric measurement methods, respectively.

The generic model has six main attributes: functionality, usability, availability, performance, maintainability, portability. Each attribute also contains a number of sub-attributes. The sub-attribute measure value will determine from the quality of their respective attributes directly.

The overall quality could be calculated through the available quality data of sub-attribute.

For the given external and internal metrics, each of its sub-attributes consists of a collection of measurement items. For example, efficiency of time attributes includes sub-attributes: response time, throughput and turnaround time. The response time metrics also constituted with two metric indexes, which is average response time and response time in the worst case. From related response time, we can found that the test is effective method to obtain the metric index value.

Here, we study the properties of all of the sub-attribute of metrics. There are three approach to get index value, which consists of obtain values from testing directly, statistics and obtain by checklist or interview.

B. Attribute Factoring

As for a given measure model, the metric attribute value need to acquire after analysis sub-attribute based on specific index value. Usually attribute computation should consider the sub-attribute and its relation between sub-attributes. In order to facilitate the calculation of index value, sub-attribute is propose to bridge the corresponding attribute measurement

 ISBN: 978-1-61208-157-1

model with metric index. Therefore, a number of sub-attributes will be measured through the composition of metric index. In some cases, metric index should be computed by several values. In these conditions, an exceed three level model is need for measurement model and its relationship between the attributes is relatively cumbersome.

If this model is analyzed with top-down method, we possible constitute measure architecture with multi-level. So the relation is relatively redundancy and increase the analyzed complicated.

This goal of research is to simplify the existing multi-level model. For reason, we adopt the idea of attributes based on the direct decomposition of GQM model [8]. The idea of GQM is contain target problem, question and measurement items.

The goal of GQM model is address the object via the question, which the question is the need to raise relevant issues for this object. Finally, measure item is the access method for sources of data directly.

Definition 1: The metric index is the smallest of factors in quality measurement model, including metric index information, metric objects, collection of metric index values, and extended attributes.

For the quality assessment model, the Φ is one attribute includes a number of sub-attributes: φ1, φ2 ... φn, which each sub-attributes of φ includes metric index ξ1, ξ2 ... ξn. The ξ is the appropriate factor for acquire data.

$$\Phi = f(\varphi_1, \varphi_2, \ldots, \varphi_n) \quad (1)$$

$$\varphi_1 = g(\xi_{11}, \xi_{12}, \ldots, \xi_{1n}) \quad (2)$$

where f, g are calculation functions corresponding to measurement model. Here index value ξ is collected from the different value. The index value can be a function point, software requirement, or even the data collection composed by above single value. The collection of single value is composite value of the point values.

Definition 2: Measure item is the smallest unit in the quality model. For each index ξ, there could be several measure item β.

For the collected data sets, usually metric index data have single or multiple item composition. The item data can be characterized by general information of specific data. The multiple item value also is vector value. Vector value cannot become calculate direct in this model and need process into scalar value by the way of average or the weighted average method. Common calculate method is the average point of the calculation, the composite weighted method. The average is calculated for all metric data collected with average method:

$$\xi = \frac{1}{i}\sum_i \beta_i \quad (3)$$

Composite weighted is a measure of all the points in accordance with the weigh distribution. Generally, assuming that the weigh value is w, and then calculated as follows:

$$\sum w_i = 1 \quad (4)$$

$$\xi = \frac{1}{i}\sum_i w_i \beta_i \quad (5)$$

The decomposition attributes optimized multi-metric evaluation system in measurement model. The decomposition attributes provided by defining a set sub-attribute to obtain relation between quality attributes and index. As a result, in order to get the ultimate quality attributes, we must decompose the quality attribute into final quantifiable evaluation values in accordance with the index of the quality attribute value through three-level model.

Three-level model consists of attribute, decompose attributes and their index, just as three level trees which shown in Figure 2. The root of tree represents the attributes, while the leaf nodes in tree are corresponding to different index. In decomposed model, two characters are differed from other generic quality model. The one is that same sub-attributes could have the several indexes. The other is, if there have two sub-attributes of the same index, the same index can be applied to multiple two sub-attributes. In Figure 2, the Metric Index 3 is applied not only for Sub-attribute 1.2 also for Sub-attribute 1.3. Therefore model can reduce the metric index use frequency and improve efficiency.

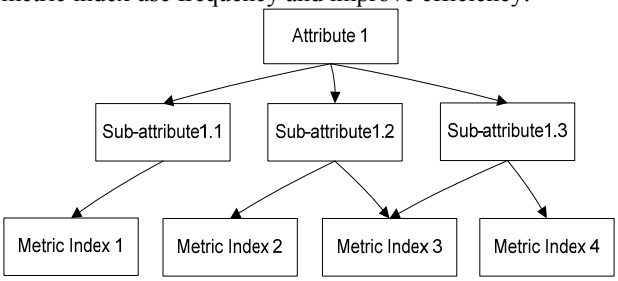

Figure 2. Direct decomposed model

III. MEASUREMENT INDEX

A. Data Collect

According to the direct decomposition model, evaluation activities need to obtain various measure items according to metric index. The item value is acquired to get the quality of data related technologies. The common technology is offline and online. Here we focus on online technology, generally real-time technology. After collecting the results of further calculations include three ways: metric index directly, function and statistics. Statistics refers to the sample on the basis of certain statistical sample. Function is the method to get metric index value through the complex calculation.

Generally, the need to obtain various types of measurement elements, namely, the direct decomposition of the metric index. The activities come mainly from the software engineering such as testing, validation and verification. The data acquisition is carried out in software engineering-related behavior of statistical analysis furthermore. Software product data adopt testing and auditing to achieve in quality model. Besides the testing, the interview is other way to obtain metric value through conducting interview.

In software quality assurance technology, testing is key approach, which can complete most acquisition of metric index value in decompose model. As far as distributed services are considered, distributed characteristics should take into account in order to obtain more accurate item data. Here the data collect is need to consider extended factors that affect the quality.

 ISBN: 978-1-61208-157-1

Definition 3: Extended factor θ is the quality of factors can influence the specific measure of certain external factors. For the different factors θ, its value will lead to measure the various different observations value y, assuming a factor of two values for x1 and x2, if x1 ≠ x2, then y1 ≠ y2, then the attribute can be considered extended factor.

For the typical extended factor, network or resource constraints is the main factor. These factors does not affect the value themselves or directly. They impact data collection of information significant in different access patterns, such as time, location, etc. Assuming the time factor t1 in day and t2 in night, obviously t1 ≠ t2, then the time duration is y1 ≠ y2 for a specific service. Therefore the time factor can be considered extended factor.

For the same measure item of quality can be used in the testing process in software product.

B. Test Collect Method

Software testing is the primary approach to obtain the testing results, which help to calculate quality value. The test approach can direct access to collected results. Here we consider the software testing, which is main approach of acquiring quality data in this study.

During software testing, test case is the basic elements during testing execution. In test collect method, a standard test case template is provided to facilitate the test execution and metric index collect. The standard test case template can assist comparative analysis in different extended factor. As for testing activities is high cost work. The test case reuse technique can improve the test efficiency and reduce cost of test.

Definition 4: Typical test case consists of test scripts, test resources, test context, test items and test.

```
<test case>::= <TCID> <Case Info>{<test item>}
<TCID>::= /*Test Case Unique ID */
<Case Info>::= <test context><test oracle><measure><test goal><test type><test method><version>
<test goal>::= product | project | technique
<test type>::= function | performance | security | others
<test method>::= manual | automated tool
<test context>::= /*test scenario for current case*/
<measure>::= <state><granularity >< reuse frequency >
< state>::= initial | modify | use | expired
```

Definition 5: Test items is single test step in one test case, including the item input, expected output, output and hint.

```
<test item>::= <TIID><item description>
<TIID>::= /*Test Item Unique ID */
< item description>::= <item input><expected output><output and hint>
<item input>::= /*test operation procedure and input information*/
<expected output>::= /*expected the result based on the input*/
<output and hint>::= /*output information and software hint information*/
```

Metric index data obtained from the testing results. In actual testing execution, the data can be get from the test results directly which defined in test results in test case.

Definition 6: Quality Data is collected through the execution of test cases.

$$d = f(tc\ o, \theta) \quad (6)$$

where tc is the collection of test cases, o is the measured object, θ is extended factor vector in specific test scenario according to test scenario.

The quality of each element can be tested to obtain the data, each element of the quality of the corresponding mapping relations between the testing requirements, so that the required index value for each measure can be tested through a series of test cases to be completed. With the metric system, the corresponding index value, by testing their statistical ways to get the corresponding index value.

Definition 7: If the same quality of data collection, its test cases is similar.

That is to satisfy for test case tc1 and tc2, if tc1 and tc2 contained same collecting metrics ξ, then tc1 and tc2 have the same measure item. If ξ (tc1)∈ξ (tc2), then tc1∈tc2, i.e., we can adopted tc1 are tc2 to complete task of data collection.

In software testing, test case design account for a large proportion. Because the test engineers need more time design test cases, reuse test case can reduce the design costs if use existing test case which have been used. It is necessary that find a common test cases in order to reduce the cost of the actual test case design after test design and test execution according to Definition 7. [10][11].

IV. DISTRIBUTED EVALUATION FRAMEWORK

The model is presented above. Here service computing has the characteristics that end user access service through any node in the network. Consider metric index value is various from the various node, which access the service due to network or time. The factors are influenced on the basis of obtain quality data using the same method from all nodes. The Session 1 shows that novel software systems are distributed mode, which different from the previous stand-alone mode. Therefore, distribution evaluation model is necessary to replace the original method that a single node, centralized evaluation of the quality metrics.

As far as service computing environment is considered, users scattered in different network node access services in the network environment. Therefore test request nodes scattered in network to replace the centralized access mode.

The traditional single-node test, this may be better for a simple without network, which results of function test are inaccuracy. But for non-functional requirements testing, such as performance, availability and reliability, the test results is affected by computer performance and configuration of computer which running the software. With the growth of network technology, the current software systems require the support of the network environment, for instance, Browser / Server or Client / Server architecture for these can be run on the network. It is worth studying in network runtime environment, especially for non-functional metric index.

Therefore, distributed test strategy can be used for accessing quality. However, many industry performance testing tools support built-in concurrent test engine to carry out the efficiency testing, which these engines often require the deployment of network-specific location. These concurrent engines do not reflect the heterogeneity of the distribution of characteristics.

 ISBN: 978-1-61208-157-1

Relative to the single-node test, the distributed testing can be close to the actual results relatively. However the distribution is relatively complex, the accuracy of data is usually due to a variety of factors, such as network structure, bandwidth, user habits and physical environment and different. In some practical applications, it can be distributed using the grid on the existence of the physical distribution. The more satisfactory distribute data could be acquired when the test case is executed in the agent on physical grid node.

A. *Evaluation System*

According to the multi-attribute decompose model and related testing techniques, we presents an evaluation system. The Figure 3 shows the architecture, as well as the main model layer structure. Quality assessment method is usually based on a large number of quality data, thus the underlying layer is responsible for access to quality data. So underlying layer is metric level to adopt testing and validate assessed object.

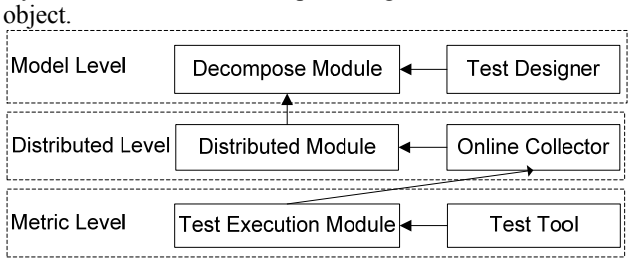

Figure 3. Evaluation System

The establishment followed the entire process from the decompose quality to generate test cases. The main steps in the process are as follows:

1) Analyzis: analyze user requirements for quality assessment and exemplify measure properties, required attribute of evaluated system.

2) Find minimum set: find the minimum measurement items with decomposed sub-attribute according to involved the metric index.

3) Test design: according to minimum measurement items, generate the test case ot reuse corresponding test case.

4) Publish: dispatch the test to various distributed access node, the node under the scenario requirements.

5) Execute: execute the test case automatically in distribute nodes, collect quality data and return to the central controller.

6) Collect data: central controller calculates the metric index value according to return quality of data.

Through the above six steps to obtain the corresponding metric index data, an online quality is available in distributed environment.

A measurement system is developed in practice. The platform concerns the design of the measurement model, in addition, the middleware in system integrated test case reuse library which support the reuse test case. Through analyzing the user's quality requirement, the system analyzes their requirement for test reuse test cases to obtain the required quality metric index data.

B. *Case Study*

Manufacture is the economy basic industries, which output required for a large number of social economic. Manufacturing information technology enhances the manufacturing and the development of integrated design, production, circulation and management efficiency. In the road of development in manufacturing industries, from the capital-intensive, technology-intensive development of the information-intensive, the computer introduces database technology, network technology, and distributed computing technology to the grid system integration technology.

Service computing environment is distributed computing. The dynamically change needs that require access to resources and services. Application and service request resources from the service rather than the traditional sense of physical entity. Service is that somehow self-maintenance and management of virtual resources. For manufacturing enterprises, the dynamics of self-maintenance and management of virtual resources in the intensive resources, utility computing, and information technology play a role in the progressive development for the enterprises are more concerned about the production, business and other technological innovations.

Figure 4 shows application integration environment. The entire application architecture is constituted with a virtual environment to achieve configuration, deployment, service. Integrated environment including enterprise, information systems and external information based on the software as a service (SaaS) model system, enterprise information system through the service encapsulation. The original model provides computing resources and storage resources, for integrating enterprise legacy systems. The enterprise information support upgrade and configure, some systems or new systems can use in the distributed environment.

The entire application environment in a typical manufacturing information applications such as CAD, CAM, CAE, PDM, ERP, CRM, SCM, MES, BI, and BPM, etc., design and manufacture of these platforms from the bottom and resource management to manufacturing execution, and then to business process reengineering and business intelligence, service environment provides the existence of information resources sharing and mutual relationship. In this environment, applications and system architecture meet the functional requirements. However the production scale, continuity, and other demand information in manufacturing environment is differ from the traditional systems. The requirement of customers with real-time, operational complexity and inclusiveness, flexibility and robust features is very high, so the overall quality of the model is concerned.

According to manufacturing information property, quality modeled at first. Not only function for a range of business system covering the production organization, production management and production optimization, but also the continuity of production to meet the requirements of non-functional metric index. And then establish manufacturing information quality model with general purpose based on the use of reusable test case in distribute environment.

 ISBN: 978-1-61208-157-1

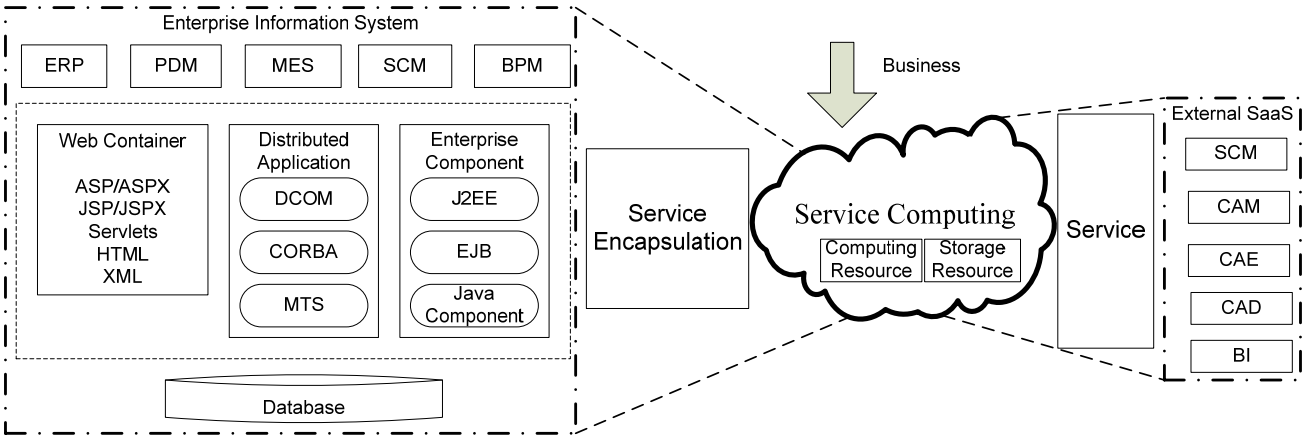

Figure 4. Typical Case with Service Computing

V. CONCLUSION AND FUTURE WORKS

In this paper, distributed quality measurement model is proposed suitable for the actual data quality during practical scenarios. The decompose quality models is key technology which goal is distribution of online access the value of the collection by running the stage. Based on distributed agents under different networks, the results of the test obtain qualitative assessment of metric values can be a more accurate in analysis system quality and runtime scenario.

Evaluation framework adopt the distributed measurement services, online execution can be more closer to the actual scenario, which not only can serve as a software engineering activities in the software online test and evaluation before online, but also provide the maintenance phase of the software to conduct regular assessment to obtain the software lifecycle quality information. The study proposes the distribution of online quality evaluation in the distributed environment for the deployment of a large user, many applications manufacturing information system and has good prospects.

The measurement model could be analyzed factors further which affecting the difference results of the final measure. On the one hand, access location of the network bandwidth will influence the evaluation results. On the other hand, the time factor is also necessary for distributed applications. With distribution of the international business applications, the peak time will occur varied in number of users and business regions.

Distributed quality evaluation is the direction of research with software technology. Future works can study related algorithms furthermore, especially evolution quality model of functions and non-functional in the system runtime and maintenance. And the benchmark library is needed to form based on numerous testing results which conform to international quality standards. In addition, the playback technology of same test cases in software regression testing could be studied for improving the authenticity of the results.

ACKNOWLEDGMENT

The work is supported by National Torch Program under Grant No. 2009GH510068 and Shanghai STCSM Program under Grant No.10DZ2291800.

The authors would like to thank without knowing the name for their helpful suggestions.

REFERENCES

[1] Chen K. and Zheng WM. Cloud computing: System instances and current research. Journal of Software, 2009, 20(5):pp.1337−1348.

[2] Bertolina A.,Angelis, De G.,Frantzen, L., Polini, A., “The PLASTIC Framework and Tools for Testing Service-Oriented Applications” Software Engineering, International Summer Schools, ISSSE 2006-2008, Salerno, Italy, pp.106-139.

[3] Offutt, J. “Quality Attributes of Web Software Applications”, IEEE Software, 2002, Vol.19(2), pp.25-32.

[4] Stephen J.H. Yang, James S.F. Hsieh, Blue C.W. Lan, Jen-Yao Chung, “Composition and Evaluation of Trustworthy Web Services”, International Journal of Web and Grid Services, 2006, 2(1), pp5-24.

[5] Boehm, B. W., Brown, J.R., Caspar, H., Lipow, M., Macleod, E. J., and Merritt, M.J. Charecteristics of Software Quality. Amsterdam: North-Holland, 1978

[6] Cavcno, J.P. and McCall, J.A. “A Framework for the measurement of software quality”. In proceedings of the Software Quality and Assurance Workshop, 1978, pp.133-140.

[7] W. Vambenepe, C. Thompson, V. Talwar, S. Rafaeli, B. Murray, D. Milojicic, S. Iyer, S, K. Farkas, M. Arlitt, “Dealing with scale and Adaptation of Global Web Services Management”, Proceedings of the IEEE International Conference on Web Services(ICWS’05) Orlando, Florida, USA, July.11-15, 2005, pp.339-346.

[8] ISO/IEC TR9126-2006, Software engineering-Product quality Part 4: Quality in use metrics

[9] Van Solingen, Rini; Egon Berghout, The Goal /Question/ Metric Method, 1999

[10] Wee Kheng Leow, Siau Cheng Khoo, Yi Sun, “Automated generation of test programs from closed specifications of classes and test cases”, Proceedings of the International Conference on Software Engineering, 2004, pp.96-105.

[11] S. Jones, “Toward an Acceptable Definition of Service”, IEEE Software, May, 2005, 22(3), pp. 87-93