International Journal of Information Technology and Business Management
29th April 2014. Vol.24 No.1
© 2012 – 2014 JITBM & ARF. All rights reserved

ISSN 2304-0777 www.jitbm.com

THE APPLICATION OF SIX SIGMA TO PROMOTE INFORMATION SYSTEM SERVICE QUALITY

Huey-Der Chu
Department of Information Technology
Takming University of Science and Technology
Taipei, Taiwan
e-mail: hdchu@takming.edu.tw

Abstract—With the increasing popularity of the Internet, Information System (IS) has become increasingly more profitable. The quality of the IS services should be confirmed prior to deploying it to the web site, resulting in customers that keep coming back. Therefore, this paper presents a framework that integrates Six Sigma Concept to improve the IS service process. The application in the selected company is a multi-tier framework system and there are 1,022 client sites distributed in Taiwan area, one broker server site and three application server sites, which concurrently access and store data from a center database server. When delivering application to the Intranet, users dissatisfaction was at an all time high. We applied the Six Sigma quality cycle: Define, Measure, Analyze, Improve and Control (DMAIC) to resolve this problem. We plotted user complaints against response time (define), drew up a test plan and executed the load testing (measure), collected testing data from different client sites for analysis (analyze), found where the bottleneck was and fixed it (improve) and monitored the status to satisfy users' requirement (control).

Keywords- Process Improvement, Six Sigma, Information System Service Quality

I. INTRODUCTION

Zona Research concluded that a majority of users would tend to abandon viewing a web site if the web page cannot be accessed in 8 seconds. Research by Robert Miller and Jakob Nielsen further indicated that most users who browse the Internet tend to spend no more than 10 seconds if it would be an interactive web page. While the profitability of commercial web sites is inextricably tied to the network traffic, an under-performing system can spell loss, for it could lead to reduced profit at the very least or a compromised business reputation at the worse scenario.

While there is virtually no geological constrain to the user thanks to the rapid development of the World Wide Web Internet as the user can be anywhere as long as they have thanks to the Internet access. The number of users also quadrupled when compared with the conventional client/Server applications, in which the predominant emphasis of a real-time interaction has also made system performance a critical issue. Since the customer is next in line, there is no room for error. When customers use that application, they do not care how the problem occurred. Instead, they just concluded that the Web site "does not work". The last thing a company wants in this seamless and competitive environment is to tarnish its image and efforts with inferior Web quality. However, one bottleneck in the Information System service remains: "How to get the deliverable right at the first time around."

Six Sigma is a quality concept made popular by Motorola's quality improvement during the 1988-1989 timeframe. Since then, it has been applied successfully to many service-based organizations [1]. When Web development and Six Sigma are used together, the result is Internet excellence [2,3,4]. Therefore, the purpose of this research is to plan Six Sigma quality into an information technology project which will add the benefit of a shared vision towards excellence and smooth transactional services.

II. SERVICE QUALITY

The SERVQUAL method from A. Parasuraman, Leonard L. Berry and Valarie A. Zeithaml (1985) [5] is a technique that can be used for performing a gap analysis of an organization's service quality performance against customer service quality needs. SERVQUAL is an empirically derived method that may be used by a services organization to improve service quality. The method involves the development of an understanding of the perceived service needs of target customers. These measured perceptions of service quality for the organization in question, are then compared against an organization that is "excellent". The resulting gap analysis may then be used as a driver for service quality improvement.

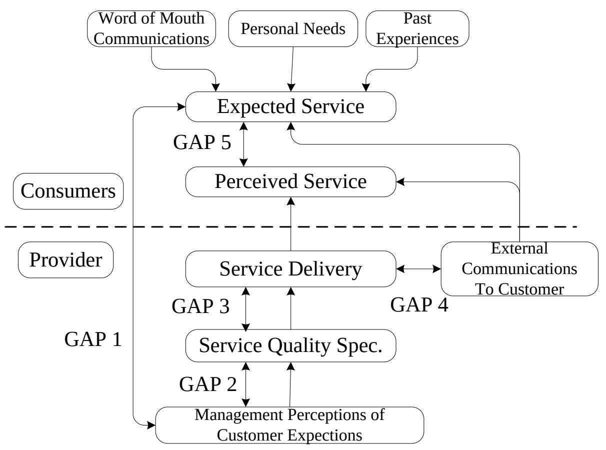
International Journal of Information Technology and Business Management
29th April 2014. Vol.24 No.1
© 2012 – 2014 JITBM & ARF. All rights reserved

ISSN 2304-0777 www.jitbm.com

This instrument showed in Figure 1 assists organizations to establish ongoing 'listening systems' to develop continuous insight about customer service needs.

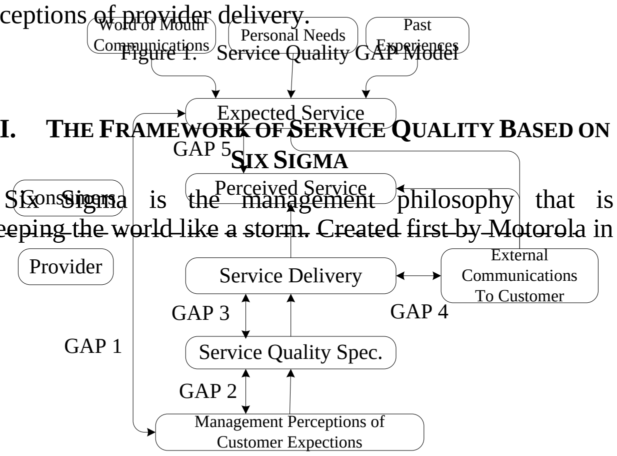
The possible contributing factors for each of the organisational Gaps are listed below. The challenge to the organisation is to isolate which variables are influencing service quality perceptions negatively and how to eliminate them. Of key importance to your organisation is Gap 1. Gap 5 relates to the overall perception your client-base has of your unit's ability to deliver on service commitments made.

Gap 1: Discrepancy between actual customer expectations and management perceptions of those expectations. Gap 2: Discrepancy between management perceptions of customer expectations and service-quality specifications. Gap 3: Discrepancy between service quality and service actually delivered. Gap 4: Discrepancy between service actually delivered and what is communicated about the service to customers. Gap 5: Discrepancy between customer's expectations of the service provider and their



perceptions of provider delivery.

Figure 1: Service Quality GAP Model



III. The Framework of Service Quality Based on Six Sigma

Six Sigma is the management philosophy that is sweeping the world like a storm. Created first by Motorola in the 1980s, then popularized by AlliedSignal and General Electric (GE) in the 1990s, Six Sigma has more than proven its worth to organizations attempting to improve their productivity and profitability [1]. It is the first and foremost a business process that enables companies to increase profits dramatically by streamlining operations, improving quality and eliminating defects or mistakes in everything a company does, from filling out purchase orders to manufacturing airplane engines. While traditional quality programs have focused on detecting and correcting defects, Six Sigma encompasses something broader: It provides specific methods to re-create the process itself so that defects are never produced in the first place.

Most companies operate at a 3-4 sigma level, where the cost of defects is roughly 20 to 30 percent of revenues [6]. By approaching Six Sigma – 3.4 defects per million opportunities – the cost of quality drops to less than 1 percent of sales. This is because the highest quality also results in the lowest costs. When GE reduced its costs from 20 percent to less than 10 percent, it saved a billion dollars in just two years—money that goes directly to the bottom line.

The method GE and several other organizations use to improve processes is summarized by the initials DMAIC as following [1]:

- Define: Defining the team to work on improvement, defining the customers of the process, their needs and requirements and creating a map of the process to be improved.
- Measure: Identifying key measures of effectiveness and efficiency and translating them into the concept of sigma.
- Analyze: Through analysis, the team can determine the causes of the problem that needs improvement.
- Improve: The sum of activities that relate to generating, selecting and implementing solutions.
- Control: Ensuring that improvement sustains over time.

Based on the concept of Six Sigma , the framework of service quality improvement is shown in Figure 2.

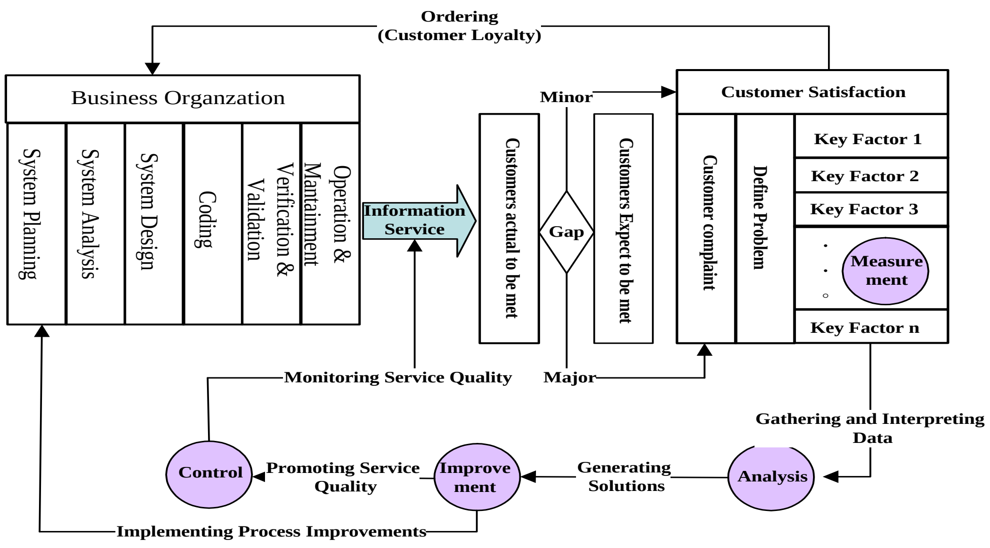


Figure 2. The framework of service quality improvement

IV. CASE STUDY

The selected company has an island-wide presence, with a total of 1,022 user points at the present time, where the merchandise of all outlets are requisitioned through the company's merchandising distribution information system, which has been streamlined toward the end of 2001. Despite an array of operating functions, the user units and company executives remain skeptic of the system's loading response performance. Bound by the pressing development timetable and the lack of auditing manpower, the absence of a complete software validation prior to the system's induction speaks of a potential concern in most system development processes. Although the chances of the firm's nationwide branches coming online simultaneously are rather slim, it is nevertheless prudent to seek remedy by rechecking the execution efficiency of the merchandising distribution information system before severe consequences may result. This has spawned the desire to have the system's loading capacity validated through professional testing before the system is fully streamlined.

The system has a three-tier framework design, divided into the users on the client end; web server, broker server and application server on the application end; database server on the server end. Currently, there are 1,022 users on the client end, one broker server and three application servers on the server end. And the user goes online through the company's Intranet or the dial-up service. The system's hierarchical framework is shown in Figure 3.

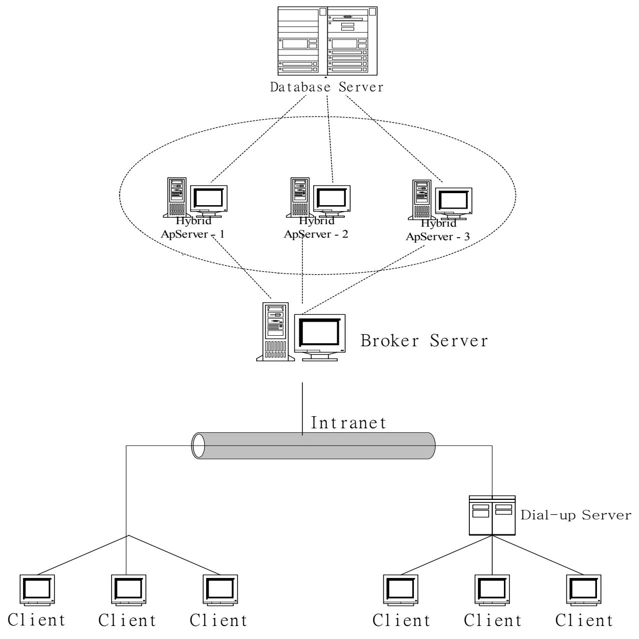


Figure 3. Diagram of the Merchandising Distribution Information System

The system operates on the WinSock protocol, instead of the more popular HTTP protocol. The user logs on to a broker server via an exclusive operating program, where the broker server routes the user's account number and password to compare them with the user's account number data retrieved from the database. The validation signals are then

International Journal of Information Technology and Business Management
29th April 2014. Vol.24 No.1
© 2012 – 2014 JITBM & ARF. All rights reserved
ISSN 2304-0777 www.jitbm.com

routed to one of the application server, where an operating program executes the request and inquiry. The web page server in the mean time supports business data inquiry and downloads operating programs for update purposes.

A. Define Phase

While company assessment estimating the request interval to business volume puts the maximum roof on simultaneous company-wide online requests at 168 users, the system loading capacity should be sufficient to accommodate the demand of 168 users simultaneously executing their online commands. Nevertheless, there is no way of knowing how the system will respond under such loading demand since the system has not yet encountered such loading demand, let alone that it may be difficult to grasp when such type of similar loading requirement will occur in the actual processing, hence the study has aimed to utilize effective testing to examine the system's response vs. its loading performance [7]. Moreover, the company also wishes to determine if the design framework of the system has been sufficient, and if there is any improvement needed to clear of any potential bottleneck from its anticipated efficiency.

Taken into account the system settings, several difficulties stand to confront the test:

- Despite there are many effective testing tools for the three-tier system out in the software market today, the system's WinSock protocol makes it difficult to locate a simpler version of the testing tools that can completely support the required test items.
- The system is best not be interrupted by the testing operations being that an already streamlined operating system needs to support service demands coming from the users across the island on a daily basis.
- Unlike what most automated testing tools taken to the capture and replay mode in executing the test, which allows a small amount of input for duplicating a large amount of test data needed to simulate a multiple number of virtual users operating online for conducting peak load testing, there are factual difficulties in adopting automated testing tools being that the system's users are assigned with fixed account number under strict clearance, and that the operating sequence being executed are limited to what has been authorized.
- It may be infeasible to measure the system's actual loading response unless the required users come online at the same time, and there are many problems to be overcome in terms of preparing the test setting, coordination of the operation, manpower and budget, before the method can be applied.

In light of the above, the test team's assessment leads to surveying for valid samples before the system's performance variation tendency can be concluded via statistical analysis, which may then be used for estimating the system's response time under a designated load; following depicts the final implementation sought.

B. Measure Phase

The project rates the system's overall response efficiency as the objectivity of the assessment, together with predefined test requirements, to come up with 100 test accounts (Test001 – Test100), in which a multi-sequencing test program has been devised by project researchers to sequentially activate the test program according to different account numbers, whereby a designated test hits are loaded to the server end, and the database is checked for inquiry update in every test cycle; its designed functions are described as follows,

a) Time synchronization is executed during the initial online linkup, designed to synchronize the timing sequence between the client end and the server end.

b) The test procedure is then executed automatically at the designated time according to the number of hits programmed.

c) Actual commands of insert, update, delete and select are simulated to the merchandising distribution information system.

d) A testing diagram, as shown in Figure 4, has six time sequencing recording points (T1 – T6) designed according to the system framework, where data on each time sequencing point occurred from T1 – T6 in each cycle are recorded in the log file.

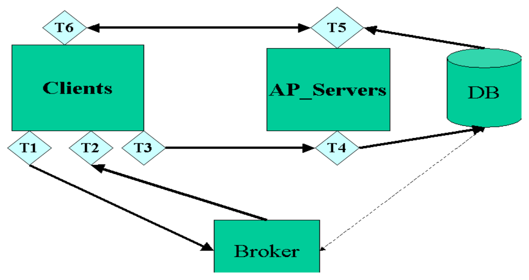


Figure 4. Testing Framework

e) Each execution program's startup time and related parameters are encoded for identification purposes. Once the program has been executed in full, pertinent data on the timing sequencing of the program ID, startup time, end time and errors are reloaded to the log file, where the log files are identified by sessions. The logging process is as follows.

- Log files are created on the client end's test computer, where timing synchronization and startup time are recorded.

- With the client end's startup time command defined as T1, the program is then linked to the broker server for identification validation, where the broker server will link to the back-end database to carry out identification validation; a correct identification will allow the broker server to assign an application server to the client end's program, and a time sequencing point T2 is recorded.
- The client end's program retrieves the business program at the application server, and a time sequencing point T3 is recorded.
- The application server then executes a business program containing SQL statements to retrieve data from the database, where the starting time being T4 and the end time being T5.
- The application server reverts the retrieved data to the client end's test program, and a sequencing time T6 is recorded.
- As soon as the data transmission is completed, the client end's program will call out a business program from the application server and reverts the application server's execution timing of T4 to T5. The multiple number of SQL statements contained in a group of programming will require the timing of each SQL statement be recorded, meaning that there will be corresponding time of T3 and T6 being recorded.
- As each transaction is completed, the response time of T1 to T6 and related data are recorded on the client end log files, being ready for subsequent analysis.

For security, a complete backup of the database server, including the operating systems, database and so forth, is run eight hours prior to the testing occurred. In support of testing the test data segments on the database server in order to run the application server under the existing system framework, the service programming contained on the application server are modified without jeopardizing the normal operations so that both the automated test program accessing the designated test data segments and the normal users accessing online data can both log on and access the system. To ensure the accuracy of testing, the client ends hardware and software equipment are kept uniformed, where the database server, broker server, application server and web server are tested using the existing servers and conducted during normal office hours between 8am and 17pm, seeking to conclude dependable ratings on the loading performance of the servers.

Given that time delay invariably occurs in every step of the program execution to and from the client server via the networking, most of the delays are created by the router's storage and transmission characteristics if the retrieval is conducted over the Internet or an Intranet, a factor that is largely dependent on the number of routers between the server and user. To best avoid the impact of such delays, the test sites are selected on those nearest the server end, together with the client ends' six test computer hardware and software setup kept uniform, to cut down the performance variations on each of the equipment which may affect the accuracy of test data concluded.

The testing procedure, sequentially carried out by the test team, has the test procedure perfected through repeated data testing and experience accumulation. While some of the test values had not been as ideal for the lack of system comprehensive in the initial stage; for example, the excessive repetitive user hits or the extended interval with increased user load would invariably lead to a surge in system response time to drag out the test schedule and result in having to abandon a test intermittently. After repeated experiments, an optimal repetitive execution on every test account has been concluded with cumulative experience indicating that an average of performance timing is at 2.251 second per hit being the best in executing the cycle of the test program.

To obtain a large sample of statistics without dragging on the test schedule due to interruption of networking or interfering the normal operations, timing interval that suffices to screen out 30 sets or more has been set under the increased system loading to allow the execution time be completed within no more than two hours as the number of users decreases. We have adopted the design of Poisson distribution model as means for evaluation a user-loading model. Eventually a 30-second interval has been used for testing the surge user model, in which six computers are used to execute the test and each test computer is to sustain six execution programming, starting from one and incrementally increased to up to thirty-six test programming as shown in Table 1, where the entire test schedule has been manipulated to run within a 2-hour span.

TABLE I. SUMMARY OF TEST ACCOUNT PLACEMENT

● **Test Computer**	● No.1	● No.2	● No.3	● No.4	● No.5	● No.6
● **Test Account**	● Test001 ● To ● Test006	● Test007 ● To ● Test012	● Test013 ● To ● Test018	● Test019 ● To ● Test024	● Test024 ● To ● Test030	● Test031 ● To ● Test036

International Journal of Information Technology and Business Management
29th April 2014. Vol.24 No.1
© 2012 – 2014 JITBM & ARF. All rights reserved
ISSN 2304-0777 www.jitbm.com

C. *Analyze Phase*

In the actual testing process as the number of users increases, the response time will also go as indicated in Figure 5 for a correlation of response time vs. user base, indicating how detailed changes occur when the system is loaded. Despite there may not be clear indication of a maximum system loading capacity, we are somehow able to establish a response tendency of a loaded system under the test mode.

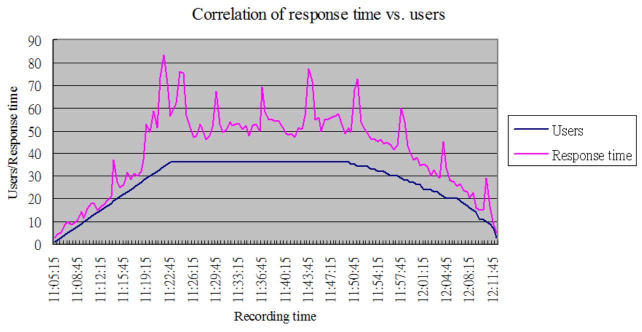


Figure 5. Comparison of Response Time vs. User Base

The test load is increased at a 30-second interval, using every 30 seconds for grouping the time sequencing data created, starting from the first test account until all test accounts are run through, which took a total of 4,056 seconds. From which, a total of 135 time segments can be derived. The Kolmogorove-Smirov statistical means concluded from the test data include a rating of 0.158189 as shown in Table II.

At a significant level where α equals to no more than 0.005, and the abandoned area being

$$C = \{d^{135} \mid d^{135} \geq 0.16081\}$$

as located from the chart, The K-S statistical means of 0.158189 that has not fall within the abandoned area of 0.16081 will not rule out the data's experience distribution as a Poisson Distribution based on the theoretic model that,

$$f(x) = \frac{e^{\lambda} \cdot \lambda^{x}}{x!} \quad , \quad x = 0,1,2,3\ldots$$

What can be concluded from the theoretic model are,

$$f(x) = \frac{(20.089)^{x} e^{-20.089}}{x!} \quad , \quad x = 0,1,2,3\ldots$$

The model's average means, λ, has been concluded at 20.089 persons per hit as the test rating, which indicates that the system is able to process 20.089 persons/hit on the online login-logout command; in other words, the average timing of online login-logout per person can be concluded at 30/20.089, or 1.4933 seconds. Given that the average response time of 1.4933 seconds has been concluded from the test, in conjunction with the company's estimated loading capacity of 168 persons to go online simultaneously, the average response time per person will roughly be at 4.18 minutes when the system is processing a 168 person load. As to whether this response time falls within a normal range, it will require the company executives to further look into the normal processing time afforded by the system design.

A further examination of the test record files indicates that no excessive variations have been found in the business processing stages of between T3 and T6. To decipher the state of changes in processing time, we further divide the transaction time into two segments – one being the log in phase, meaning that covering T1 to T2, and the other being the business processing phase, meaning T3 to T6, where the sum of both will provide the transaction time of a given test. While plotting a total of 2,715 of the processing time and total transaction time concluded from the two stages as shown in Fig. 4 indicates that the longer the transaction is dragged on, the more ominous the login time, the yellow zone, will become, but variations in the business processing time, the red zone, remains unchanged. This clearly indicates that the system's login processing sequence is ominously reduced of its processing speed when the load increases, an eventual bottle neck when the system is fully streamlined and an area that calls for system developers to further examine it for viable solution.

TABLE II. THE KOLMOGOROVE-SMIROV STATISTICAL MEANS CONCLUDED FROM THE TEST DATA

Class Groups of values	$F_{actual}(x)$ Actual value	$F_{expected}(x)$ Expected value	$\lvert F_{actual}(x) - F_{expected}(x) \rvert$
0~2	0.000000	0.000000	0.000000
3~4	0.014815	0.000016	0.014799
5~6	0.029630	0.000239	0.029391
7~8	0.096296	0.001974	0.094332
9~10	0.133333	0.010306	0.123027
11~12	0.192593	0.037473	0.155120
13~14	0.251852	0.101467	0.150385
15~16	0.311111	0.215386	0.095725
17~18	0.370370	0.373954	0.003584
19~20	0.451852	0.551197	0.099345
21~22	0.555556	0.713745	**0.158189**
23~24	0.711111	0.838229	0.127118
25~26	0.807407	0.919024	0.111617
27~28	0.881481	0.964025	0.134978
29~30	0.933333	0.985767	0.052434
31~32	0.992593	0.994965	0.002372
32~34	0.992593	0.998401	0.005808
34~	1.000000	1.000000	0.000000

International Journal of Information Technology and Business Management
29th April 2014. Vol.24 No.1
© 2012 – 2014 JITBM & ARF. All rights reserved

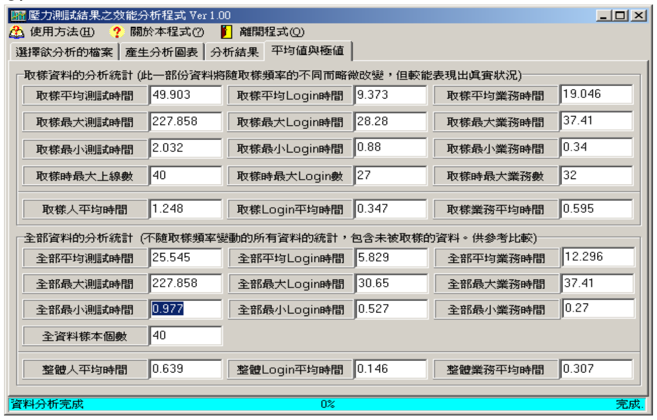
ISSN 2304-0777 www.jitbm.com

D. Improve Phase

The example that a loading test can be completed without being aided by any of the automated testing tools being available on the market, and have the test objectives were achieved, does allow company executives to review estimated ratings of a response time, while the bottleneck created by the system's login time, between T1 and T2, remains the focal point of improvement in this phase. The system developer has already begun to modify the system framework by adding the number of broker servers, modifying the login module on the client end for an interactive selection of broker servers, switching the account number validation function to under the application server for execution, to address concerns of bottleneck in system login time.

E. Control Phase

In this case study the team wanted to be sure that the improvements, once implemented, held value and did not revert to error-riddled baseline. The team modified the test framework to be a monitor to watch the application performance whether or not it satisfies users requirement. The tool for monitoring the application is as shown in Figure 6:



取樣資料的分析統計 (此一部份資料將隨取樣頻率的不同而略微改變，但較能表現出真實狀況)

取樣平均測試時間	49.903	取樣平均Login時間	9.373	取樣平均業務時間	19.046
取樣最大測試時間	227.858	取樣最大Login時間	28.28	取樣最大業務時間	37.41
取樣最小測試時間	2.032	取樣最小Login時間	0.88	取樣最小業務時間	0.34
取樣時最大上線數	40	取樣時最大Login數	27	取樣時最大業務數	32
取樣人平均時間	1.248	取樣Login平均時間	0.347	取樣業務平均時間	0.595

全部資料的分析統計 (不隨取樣頻率變動的所有資料的統計，包含未被取樣的資料。供參考比較)

全部平均測試時間	25.545	全部平均Login時間	5.829	全部平均業務時間	12.296
全部最大測試時間	227.858	全部最大Login時間	30.65	全部最大業務時間	37.41
全部最小測試時間	0.977	全部最小Login時間	0.527	全部最小業務時間	0.27
全資料樣本個數	40				
整體人平均時間	0.639	整體Login平均時間	0.146	整體業務平均時間	0.307

資料分析完成 0% 完成.

Figure 6. Tool for monitoring the application

V. CONCLUSION

Management problems in the information system service have been addressed over the last years by a strong focus on the improvement of the service processes. Six Sigma is the most powerful breakthrough management tool ever devised, promising increased market share, cost reductions and dramatic improvements in bottom-line profitability for companies of any size. The method of Six Sigma mapped the service process for information system against the Six Sigma quality cycle (DMAIC): Define, Measure, Analyze, Improve and Control and then aligned those cycles against the quality management life cycle (PDCA): Plan, Do, Check and Action. The case shown in this paper applied Six Sigma to reduce the Gap 5 and improve the quality of service process. It demonstrated that this method not only assisted the development team to focus on reducing service variation, but also gave the rest of the company the understanding to the IT involvement necessary for success.

REFERENCES

[1] Eckes, G. (2001), Making Six Sigma Last—Managing the Balance Between Cultural and Technical Change, John Wiley & Sons, Inc., USA.

[2] Harrington, H.J. and McNellis, T. (2001), "Six Sigma for Internet Application Development", Software Quality Professional, 4(1), pp. 7-18.

[3] Mahanti, R. (2005), "Six Sigma for Software", Software Quality Professional, 8(1), pp. 12-26.

[4] Siviy, J., Stoddard, R. and Penn, M.L. (2007), "Achieving Success via Multi-Model Process Improvement Program", SEPG 2007 conference, Austin, TX, USA

[5] Parasuraman,A.;Berry,L.L.;Zeithaml,V. A. (1985), "A Conceptual Model of Service Quality and Its Implications for Future Research", Journal of Marketing, 49(4), pp.41-50.

[6] Harry, M and Schroeder, R. (2000), Six Sigma—The Breakthrough Management Strategy Revolutionizing the World's Top Corporations, Random House, Inc.,USA.

[7] Nguyen, H.Q., (2000), Testing Application on the Web, John Wiley & Sons, Inc.,USA.