CMG JOURNAL

2017 | ISSUE 3

THIS ISSUE:

CMG
Computer Measurement Group

ManagingEditor: Stephen Guendert, Ph.D.
Issue Editor - Sarah Guendert

The Computer Measurement Group, Inc. (CMG) is the most influential organization worldwide for the exchange of information among Computer Performance Evaluation (CPE) professionals. To keep CPE professionals on top of their field, CMG is dedicated to providing its members the education, networking, and leadership opportunities vital to success in today's competitive industry.

As a benefit to its members, CMG is pleased to publish this CMG Journal and it complimentary publications, the Proceedings, Amplify Blog, and MeasureIT Newsletter.

For more information about CMG and the benefits of membership, please visit www.cmg.org.

INFORMATION FOR AUTHORS

The Journal of Computer Resource Management (CMG Journal) is distributed only to members of the Computer Measurement Group, Inc. (CMG). CMG membership consists primarily of practicing professionals in the area of computer performance, analysis, and capacity planning. The objective of the CMG Journal is to bring the membership information of current importance involving:

- performance characteristics of computer systems of all sizes and architectures
- techniques to measure, analyze, compare, predict, and report performance
- techniques for managing computer performance, including capacity management, computer cost accounting, and computer performance reporting
- integration of disparate components into hardware and software systems to meet organization needs.

The intent of CMG is to publish relevant papers as rapidly as possible, preferably within six months of completion. Authors can assist CMG in achieving this objective by submitting papers in a uniform format.

To submit your paper for consideration in the CMG Journal, it must follow a format consistent with CMG guidelines. For complete instructions on paper submittal you should download two documents from https://www.cmg.org/publications/cmg-journal/ located on the CMG Website. These documents are 'Paper Guidelines and Instructions' and 'CMG License to Publish.' The first document will provide guidelines on format and text; the second document is a required form that must be signed/returned to CMG that grants your permission for CMG to publish your paper.

Here are some highlights on what is expected from authors:

- Camera-Ready Papers, in Microsoft Word, should be sent via email to cmgjournal@cmg.org
- Electronic copies of the paper are preferred; send two hardcopies of the paper in the required format (see 'Instructions' document for details) if an electronic copy is not possible. Hardcopies must be camera ready Papers should be submitted three months prior to the Journal publication date
- Please indicate if you wish to review the paper after editing, prior to publication
- Refer to all tables and figures by number

Authors must read and follow the 'Instructions for Preparing Papers' document. The above highlights do not include all the guidelines required for submitting your paper. Papers not complying with the instructions may be rejected for inclusion in the CMG Journal.

Published by the Computer Measurement Group, a not for profit Illinois membership corporation. Publication in the CMG Journal implies acknowledgment of the author's (or authors') copyright and republication rights. Permission to reprint in whole or in part may be granted for educational and scientific purposes upon written application to the Editor, the Computer Measurement Group, 151 Fries Mill Road, Executive Campus, Suite 104, Turnersville, NJ 08012. Permission is hereby granted to members of CMG to reproduce this publication in whole or in part solely for internal distributions within a member's organization provided the copyright notice printed above is set forth in full
text on the title page of each item reproduced. CMG acknowledges the ownership of trademarks and registered trademarks that appear in this publication. They are each to be regarded as appearing with the appropriate ® or ™ symbols at first mention.

The ideas and concepts set forth in this publication are solely those of the respective authors, and not those of CMG. CMG does not endorse, guarantee, or otherwise certify any such ideas or concepts in any application or usage.

Journal of Computer Resource Management

A Publication of the Computer Measurement Group

2017 Issue 2

Table of Contents

CMG Journal 2017 Issue #3

Letter from the Editor

WELCOME to the third issue of CMG Journal for 2017. It’s hard to believe that CMG Impact 2017 is just a short 6 weeks away. What better way to get the brain warmed up for the great conference ahead than by reading this, the latest issue of CMG Journal. This month we have three papers for your enjoyment.

Our first paper was written by Ashutosh Shinde, *Selecting Enterprise Applications for Performance Audit Using Analytical Hierarchy Process (AHP),* discusses a technique for introducing objectivity and dealing with a complex decision-making process while selecting the application set. The paper applies the technique to a typical banking enterprise and explains how the critical applications can be selected.

Our second paper, *Working on Application Slowness in Mainframe Infrastructure-Database Recommendations,* written by Venkatesh Pandian Rajagopalan, aims at addressing the capacity and performance issues happened in an infrastructure project and the various methods considered to overcome the performance bottleneck in a mainframe application and infrastructure.

Last, but certainly not least, our third paper, *Re-inventing ICT but Not the Wheel,* was written by Flavio Gaj and Silvio Antonio Varagnolo. Their paper discusses the concept of ICT Service Management” (Information and Communication Technology Service Management), along with reference models, guidelines and certification systems that are part of this discipline. Of the reference models, the Information Technology Infrastructure Library (ITIL) is the most consolidated, exhaustive and closest to the problems of ICT Service Management.

We are actively soliciting papers for future issues of CMG Journal. We are planning on publishing four issues per year, with a goal of four to five high quality papers per issue. Our next issue is planned for December 2017 and we are currently looking for and reviewing submissions. Please consider writing a paper for the CMG Journal. You can submit your papers, as well as feedback to us at cmgjournal@cmg.org.

Thanks again for reading, we hope you enjoy this issue and hope to see you at CMG Impact 2017.

Stephen R. Guendert, Ph.D.

Selecting Enterprise Applications for Performance Audit Using Analytical Hierarchy Process (AHP)

Ashutosh Shinde
ashutoshs2017@gmail.com

Abstract—**Performance and scalability issues in Enterprise applications are leading to multi-million dollar losses to Businesses across the globe. Such issues lead to bad customer experiences, lower employee productivity, reduced operational efficiency and ultimately impact the business. Regular performance audit of the IT systems across the entire technology stack is a proven method to identify and fix the pending and chronic performance issues, thereby improving performance. This complements other methods like performance monitoring and testing. A large enterprise can have thousands of applications and identifying the critical applications to undergo regular performance audit can be challenging, particularly in enterprises that have many teams. Selecting the right application set can be a subjective, unstructured and error prone process. Wrong selection can lead to sub-optimal coverage and no reduction in the risk profile of the applications that matter the most. How should enterprises address this challenge and objectively select an optimal set of applications for the audit activity? Analytical Hierarchy Process (AHP) is discussed in this paper as a technique for introducing objectivity and dealing with the complex decision-making process while selecting the application set. The paper applies the technique to a typical banking enterprise and explains how the critical applications can be selected.**

Keywords—Analytical Hierarchy Process, AHP, Performance Engineering, Performance Audit

I. Introduction

Performance and scalability issues in enterprise applications are leading to multi-million dollar losses to Businesses across the globe, as the IT systems do not scale in alignment with the business demands. Customer experience, user productivity, operational efficiency, and hence overall business is affected when IT applications have performance and scalability issues. Pro-actively and regularly reviewing the applications in production to identify and resolve the performance/ scalability issues across the entire technology stack is an excellent proven practice to address the impending issues before they manifest. Such a practice also helps in resolving some of the chronic issues that lead to consistent failures. This activity is referred as “Performance Audit” for the remainder of this paper. This technique is complementary to the widely-adopted techniques like performance monitoring and performance testing. Monitoring is mostly associated with identifying and reporting the service level agreement (SLA) breaches; in comparison, the performance audit cycles are expected to identify the exact causes of the consistent breaches or failures and prevent subsequent incidents. During an audit cycle, a team of performance experts collect data like system utilization, database reports, JVM heap dumps and thread dumps, network statistics etc. and conduct a thorough health check of all the components of the application such as the web server, application server, database, network, hardware, and operating system. Solutions for the identified issues are shared with the relevant teams and product owners.

Today it is not uncommon for medium and large enterprises to have a few hundred applications in their IT system. Choosing the right applications from this large set is critical to get maximum coverage and reduce the risks through the audit activity. In an organization with multiple teams, identification of the critical application set can become a subjective process, creating a need to introduce a level of objectivity in the selection process. Analytic Hierarchy Process (AHP) [1] by Thomas Saaty is a structured technique for dealing with complex decisions and relies on the judgment of experts to derive priority scales. It is a heuristic algorithm which allows inclusion of a certain degree of objectivity in an otherwise complex subjective selection process.

This mathematical decision-making model is explored in this paper as a technique to identify the right set of applications as candidates for performance audit and optimization activity. The outcome helps in prioritizing the order in which the applications must go through the audit and optimization cycle. Enterprises can also define the audit frequency once a critical set of applications is identified. For example, a core banking system, which is one of the most important IT applications in a bank, may undergo an audit twice a year whereas a reporting application may be only once a year.

II. Analytical Hierarchy Process (AHP)

The AHP process is detailed in this section and is applied to the problem domain in the subsequent section. To generate the application priorities, we need to decompose the decision into the following steps:

Step 1: Model the problem as a hierarchy of goals, alternatives for reaching the goals and criteria for evaluating the alternatives. The final goal is to rank the various applications for performance audit based on various business and technical parameters which form the criteria. Enterprises can choose the parameter vector that best defines the criticality of their application in their business domain.

Step 2: Make a series of judgments to establish the weight of the various criteria using pair wise comparisons amongst the elements of the hierarchy. Each judgment is assigned a number on a scale. It is a good practice to setup a team with representatives from various functions within the enterprise, including business units, to define and then compare the various criteria. The pair-wise comparison uses the following scale.

Intensity of importance	Definition	Explanation
1	Equal importance	Two criteria contribute equally
3	Moderate importance	Experience and judgment slightly favors one over the other
5	Strong importance	Experience and judgment strongly favors one over the other
7	Very Strong importance	A criterion is favored strongly over another
9	Extreme importance	Evidence favoring one criterion over another is of the highest possible order
2,4,6,8	Intermediate values	

Step 3: The matrix created is used to calculate the relative ranking of the various criteria by using Eigen vectors [2]. The pairwise comparison matrix is successively raised to higher powers that are squared each time. The matrix is then normalized to arrive at weights for each criterion. A consistency ratio is calculated to check the consistency of the judgment.

Step 4: Similarly, a pair wise comparison matrix is created for each alternative under each criterion. Eigen vectors are calculated from the matrix which determine the ranking of each alternative under each criterion. Since the number of applications can be quite large in an enterprise it may be difficult to create a pair wise comparison matrix for every application under each criterion. An alternative method of creating a ranking scale is proposed in this paper to reduce the complexity introduced due to the large number of comparisons.

Step 5: The weight of each criteria is used to develop the ratings for each decision. Finally, we multiply the criteria ranks with the alternative ranks to arrive at the final set of weights for each alternative. The AHP technique therefore also helps in ranking the alternatives. Enterprises can then choose the top 10-20% of the applications for audit and optimization activity.

The output of this method is a list of applications ranked per their weights.

III. Case study: application of AHP in a bank to identify the critical application set

A large bank with the following applications was considering demonstrating the use of AHP in identifying and prioritizing the application list. These are common applications found in most commercial banks around the globe.

1. Core banking: maintains the accounts, balances, ledger etc. and forms the heart of a banking system, as it enables the customers to conduct their business with the bank.

2. Internet banking or E-banking: an important digital channel that offers online access to accounts and enables users to conduct a range of financial transactions over the internet

3. CRM: used to manage customers' relationships with the bank through branches and call centers. CRM in many cases is integrated with the call center software. Some applications like Internet banking and mobile banking are also channels for self-service requests.

4. Corporate Internet Banking: provides a Business-to-Business E-channel to corporations for salary payments, deposits, tax payments etc.

5. Payment gateway: provides capabilities to merchants for handling credit, debit, and direct to bank payment transactions

6. Mobile banking: Mobile App and supporting back end application to provide access to the account and transactions like money transfer through the mobile devices.

7. Account opening (mobility based): used by bank agents to open and service accounts in a digital manner by using the mobile functions like scanning for higher accuracy and speed.

8. Online Remittances

9. SMS Application to send alerts to users. Note that some countries have strict guidelines framed by regulators to deliver messages within a stipulated time frame for financial transactions.

10. Private banking: used for managing wealth portfolios typically for high net-worth Individuals

Step 1:

The criteria used for ranking the applications is below. This list can be expanded to add more criteria based on the nature of business and importance of various non-functional aspects.

A. Business value: defines the importance of the application to the core business of the enterprise. In its simplest form, business value can be considered as the revenue generated directly through the channel or the application. In another form, it could be the transaction volume serviced by the application. It can also be considered as the importance of the channel to the brand value, although the application may not necessarily be generating revenue. For example, a website created for promoting a new product.

B. Customer experience (response time criticality): Many enterprises consider customer experience as a differentiator to outshine the competition. As many studies have proven, response time is a critical factor that impacts the customer experience and poor customer experience leads to a decrease in business. Performance audit exercise is focused on identifying and addressing the bottlenecks across the technology stack, thus improving response time. Digital channels that offer a conduit for communicating with customers are expected to deliver a good customer experience.

C. Employee productivity: performance/ scalability issues in some of the applications like CRM can directly impact the productivity level of the employees. This can in turn impact the customer service offered to the end users and reduce the turnover and corporate performance. In some countries, this can lead to labor issues and legal complications if the productivity degradation leads to working overtime by employees.

D. Dependency of other applications: Performance issues in downstream applications can create a domino effect on dependent applications. For example, degradation in core banking systems impact all other dependent digital channels. Hence, identifying and pro-actively improving/maintaining performance of such applications is critical in reducing the risk to the enterprise. Technical architecture solutions also need to be reviewed and evaluated during the audit to reduce or eliminate the impact of the slowness on the upstream applications.

E. Compliance with regulators / security risk: Performance issues in systems like anti-money laundering can lead to non-compliance issues with regulators if reports are not generated and delivered on time. Many risk analysis systems are real time and hence response time sensitive. Performance issues in these systems can expose the business and its customers to unacceptable risks.

Other filters that necessitate an audit can also be added to the selection criteria. It is advisable to keep the number of criteria between 5 and 9 [4]. Some other filters are listed below:

- Large number of outstanding performance issues in existing applications that impact business.
- Applications in which future workload changes are anticipated by business that may impact performance.
- Negative sentiments in social media about performance issues in customer facing applications.
- Upcoming marketing campaigns that may increase load on some systems.
- Mergers/ acquisitions that may change the workload profile of the application etc.
- Technology/ architecture transformations. Such major initiatives mandate the need for performance baselining before the changes are implemented. This ensures there is a baseline for comparing the performance of the new and older systems.

Step 2: Make a series of judgements using pair wise comparisons of the various criteria to calculate the relative rankings.

A pair wise comparison matrix with nXn dimensions is created where n is the number of evaluation criteria. Each entry n_{ij} represents the importance of the ith element with the jth element of the matrix. Likewise, n_{ji} represents the reciprocal value of the comparison. The diagonal elements are all 1. A pair wise comparison matrix for the criteria list that we defined in the above section is created here:

	Business value	Customer Experience	Employee productivity	Dependency	Compliance / Risk
Business value	1	3	7	5	4
Customer Experience	0.33	1	6	4	0.5
Employee productivity	0.14	0.17	1	0.33	0.25
Dependency	0.2	0.25	3	1	0.25
Compliance/ Risk	0.25	2	4	4	1

Step 3: Establish the weight and rank for the criteria. An Eigen weights vector (also called as priority vector) is created to establish the weight and rank for the criteria. Web based AHP Calculators [3] as well as Excel worksheets are available for the calculations. For the above matrix, the resulting weights and rank for the criteria are shown below.

Criteria		Priority	Rank
1	Business Value	48.00%	1
2	Customer Experience	18.60%	3
3	Employee Productivity	4.10%	5
4	Dependency	7.40%	4
5	Compliance/Risk	21.90%	2

Principal Eigen value [5] = 5.352. The consistency ratio is around 7.8% which is under the generally acceptable limit of 10% [6]. The judgement needs to be revised if the consistency ratio is more than 10%.

Step 4: As per AHP, a pair wise comparison matrix is created for each alternative under each criterion. Since the number of alternative applications being considered can be sizeable in a large enterprise it may not be easy to compare applications for each criterion. Hence the alternatives are ranked on a scale of low-medium-high for each of the criterion. A pair-wise comparison matrix of the low-medium-high scale is created using Eigen vectors to determine the weight of each element in the scale. The scale can be extended to have multiple intermediate values for ease of comparison.

	High	Medium	Low
High	1	4	8
Medium	1/4	1	5
Low	1/8	1/5	1

For the above matrix, the resulting weights for the criteria are:

Criterion		Priority	Rank
1	High	69.9%	1
2	Medium	23.7%	2
3	Low	6.4%	3

Principal Eigen value = 3.094, Consistency ratio = 9.8%

The scale is now used to compare the applications for each criterion.

	Business value	Customer experience	Employee Productivity	Dependency	Compliance/ Risk
Core banking	High	High	Medium	High	Medium
Internet banking	High	High	Low	Low	Medium
CRM	Low	Medium	High	Low	Low
Corporate Internet Banking	High	Medium	Low	Low	Medium
Payment Gateway	Medium	High	Low	High	Medium
Mobile banking	Medium	High	Low	Low	Medium
Acc opening	Low	Medium	Medium	Low	Low
Online remittance	Medium	High	Low	Low	High
SMS application	Low	High	Low	Medium	Low
Private banking	Medium	High	Low	Low	Medium

Step 5: The priority weight for the criterion from Step 3 is multiplied by the priority weight for the high-medium-low scale calculated in Step 4 to determine the application score for each criterion-application pair.

Score $_{\text{Core Banking}}$ = (Priority Weight $_{\text{Business Value}}$ X Priority Weight $_{\text{High}}$) + (Priority Weight $_{\text{Customer Experience}}$ X Priority Weight $_{\text{High}}$)+ (Priority Weight $_{\text{Employee Productivity}}$ X Priority Weight $_{\text{Medium}}$)+ (Priority Weight $_{\text{Dependency}}$ X Priority Weight $_{\text{High}}$) + (Priority Weight $_{\text{Compliance Risk}}$ X Priority Weight $_{\text{Medium}}$)

The scores for all applications are added to calculate the total application score.

Relative weight is calculated for each application by dividing the application score by the total score across the applications. Weights are used to determine the relative ranking of every application in the set.

Application	**Relative score**	**Relative ranking**
Core banking	0.19429085	1
Internet banking	0.149933175	2
Corporate Internet Banking	0.12538262	3
Online remittance	0.115483203	4
Payment Gateway	0.100001828	5
Mobile banking	0.086576904	6
Private banking	0.086576904	7
SMS application	0.055685865	8
CRM	0.045740165	9
Account opening	0.040328484	10

The applications can be segregated across various action segments as shown below based on their ranking. The frequency of performance audit and performance benchmarking can be higher for the top slots whereas rest of the applications may only be monitored daily for key health parameters like CPU and memory utilization.

	Bi-annual Performance Audit	Annual Performance Audit	Annual Performance benchmarking	High-level Performance Monitoring
Top 10%	X		X	X
11-20%		X	X	X
21-30%			X	X
Rest				X

IV. CONCLUSION

Performance audit involves analysis of the entire technology stack including the hardware, operating system, application code, configurations, database performance etc. and hence is a labor-intensive activity. Each audit can consume 4-10 weeks with multiple performance architects based on the size and complexity of the application technology stack. The advantages of such an activity are immense in reducing the risk to business. However, selecting the right set of applications for the audit exercise is critical for optimal coverage and for reducing the risks. AHP provides a structured approach for dealing with complex decisions and can be adopted by enterprises as the first step in filtering the application set. Some selection criterion are described in the paper but they can be extended based on the business needs. Experiential judgement within the team can be added as the ultimate step before finalizing the list.

V. References

[1] Saaty, T.L. (2008) 'Decision making with the analytic hierarchy process', Int. J. Services Sciences, Vol. 1, No. 1, pp.83–98.

[2] An Illustrated Guide to the Analytical Hierarchy Process by Dr. Rainer Haas and Dr. Oliver Meixner - https://mi.boku.ac.at/ahp/ahptutorial.pdf

[3] BPMSG Free Web based AHP Calculator for priority calculations - http://bpmsg.com/academic/ahp_calc.php [Accessed 28-June-2017]

[4] Saaty, T.L. and Ozdemir, M.S. Why the Magic Number Seven Plus or Minus Two, *Mathematical and Computer Modelling*, 2003, vol. 38, pp. 233-244

[5] Saaty, T.L. (2002) Decision Making with the AHP – Why is the principal eigenvector necessary. European Journal of Operation Research 145 (2003) 85-91

[6] Teknomo, Kardi. Analytic Hierarchy Process Tutorial. Website - http://people.revoledu.com/kardi/tutorial/AHP [Accessed 25-June-2017]

Working on Application Slowness in Mainframe Infrastructure-Database Recommendations

By Venkatesh Pandian Rajagopalan

Email - Venkatesh_r20@infosys.com

This Paper aims at addressing the capacity and performance issues happened in an infrastructure project and the various methods considered to overcome the performance bottle neck. The underlying application and infrastructure was mainframe.

ABSTRACT:

This paper discusses the real-time infrastructure issue which was faced by an energy utility client. The application was primarily based on a mainframe environment with DB2 as its back-end and CICS as the online transaction processing (OLTP) application.

The contention issue occurred only during the peak utilization of CPU (month end processing) and application performance issues caused an outage to the business. During these periods, the project bought in proactive monitoring tools, automation scripts that will trigger alerts for the DB2 and the CICS systems, so the performance bottlenecks were eliminated before escalation occurred. Application performance tuning was studied and completed during the time of the problem. A detailed analysis of the DB2 subsystem was done and a set of tools were deployed to increase application program throughput to DB2. This alleviated the necessity of upgrading the processor in near time, resulting in cost savings for the customer. It also eliminated the necessity of entirely transforming the mainframe system.

Overall components related to application programs, database and online transaction processing were constructively studied and a much-needed change management process was implemented to overcome the crisis and help ensure business continuity.

The recommendations listed below were provided by the infrastructure team and implemented by the application team and infrastructure teams in a short time span when the crisis started. As a result, the critical incidents during this critical time were drastically reduced in a span of two months. In addition, the performance problem was arrested at the same point in time. Over 50% of P1 and P2 incidents were reduced because of the recommendations and work-around.

The paper will discuss the best practices and the proactive approach undertaken which helped overcome the performance issues in the application and in the infrastructure.

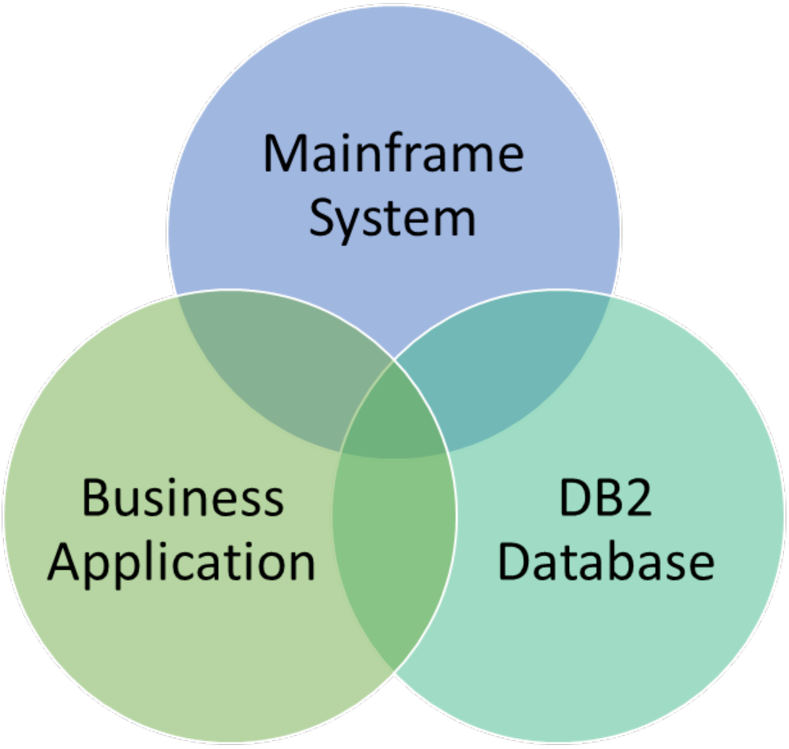
Problem Statement:

When the customer care application reported excess latency in the application front-end side, the primary DB2 subsystem was having contention with the number of threads and increasing CPU utilization. This resulted in the high number of DB2 threads coming from the OLTP system. As a result, the CPU utilization hit 99% for the production LPAR.

Due to this, most of the activities, testing, and processing in the LPAR went idle and overall performance was drastically low. As a result, business critical applications residing on the mainframe crashed and intermittent outages occurred

This typically happens when there is a heavy workload expected during month end, and financial close period processing. The capacity and performance analysis were carried out by the infrastructure team, in association with the application team.

The Approach:



To start investigating the performance issues, we understood that the problem could be with the business application (as a result of huge volume of ad-hoc data or scheduling jobs in an inefficient manner, or the DB2 data store or the mainframe system itself. The problem would likely be discovered in any of these three elements. The problem could also be multi-faceted. One can always start to investigate from these 3 levels in a mainframe environment. We can classify the recommendations and measurements to DB2 related and Non-DB2 related in our approach.

DB2 Recommendations and Implementations:

1. Queries consuming high MSUs

Programs/Packages consuming more MSU's were extracted the following day, using an Omegamon -SAS program and sent to the application team for further analysis. This gave the application team the scope to understand which programs result in costly utilization of the CPU. Post- analysis, the performance tuning of application programs-DB2 Plan was kicked off, to be optimised for better performing access path.

A sample report that was sent to the application team:

Parameter	Programs consuming high MSUs
Observation	• There are packages which consume high MSUs. Example below, PROGRAMS MSU’s (High to Low) KIDCAB0 63.47 KILIFG4 63.44 KIDSA41 45.44 KIBRFF1 37.42 KIDCAB2 32.35
Recommendation	• DB2 database team can generate a daily report on packages that consume high MSU. • Application teams should analyze the packages that consume high MSU and identify possibilities to reduce MSUs consumed by them.

2. Monitoring the DB2 Thread levels for criticality

The number of DB2 threads which were escalated from few to hundreds or even thousands was an early sign of getting in to a critical situation. As a result, the infrastructure team devised an automated method to capture the number of DB2 threads at regular intervals. The thread was also actively monitored using the OMEGAMON for DB2 tool, which was a manual and time consuming task.

Using REXX/JCL programs, we designed an automated solution to capture the number of threads in 5-10 minute intervals. If the thread count is more (>100) than the usual level of the number of threads, details of the threads segregated according to type were sent to the application team. The application team immediately analyzes these threads and takes necessary actions before the issue escalates to a critical incident. The automated approach tremendously helped in warning the team to take precautionary measures.

No manual intervention is needed as the trigger was through TWS (Tivoli Workstation)

The job will take care of the following,

- Generating DB2 threads/plans executing in DBE1 at that point in time, along with the count of threads.
- Measure the CPU% of DBE1DBM1 & DBE1MSTR address spaces
- Email the thread count, plan names and CPU details in a single mail to the recipient/group owning the application.

A sample Threshold levels for the thread monitoring job is provided below:

Threshold Levels		
Critical	**Major**	**Minor**
Add Critical Threshold Level (% or Number)	*Add Major Threshold Level (% or Number)*	*Add Minor Threshold Level (% or Number)*
1)Number of threads > 400 2)DBE1MSTR & DBE1DBM1>95% 3)SYSE CPU > 95%	1)Number of threads > 200 2)DBE1MSTR & DBE1DBM1>65% 3)SYSE CPU > 80%	1)Number of threads > 100 2)DBE1MSTR & DBE1DBM1>50% 3)SYSE CPU > 65%

3. Narrowing down the high intensive and costly queries

Queries were submitted by the application users or end users using the tool called AQT (advance query tool). The queries constructed using this tool were very large and costly queries, which used to run from several minutes to a few hours.

The following guidelines were prescribed and followed when any significant amount of processing is expected or when we sensed a critical problem is fast approaching, in terms of growing thread numbers and high CPU cost escalation.

Parameter	AQT Queries (Adhoc Queries)
Observation	• The AQT queries are run during business hours or while the batches are running
Recommendation	• Do not run AQT queries during critical periods like Monthly Close etc. • In unavoidable situations, run ad hoc AQT queries only under tight control • Schedule the static AQT queries during off business hours using a Scheduler • Make use of MQT (Materialized Query Tables) for static queries so that the query response is fast and query execution wont execute on the base table

In this line MQT (materialized query tables) were newly built in the application for the queries which are huge/complex and frequently used. This saved a lot of processing as the query was pre-built and data throughput was consistent. The number of times the MQT was accessed began to grow significantly during critical processing.

The AQT queries which were long running also posed a risk of using a lot of processor. There was an idea to find the long running query from the DB2 accounting report shared with the application users and business end users, so that they could make corrections to submit sensible queries over the time period. In this way, some of the mischievous queries were stopped from slowly getting in to the system.

Procedure to find the long running AQT queries from the DB2 Accounting Report

We can find the thread which takes more CPU and run time using the DB2 accounting report. We have the DB2 SMF Records (100,101,102) routed to the following daily basis SMF dataset HLQ for all the DB2 subsystems.

SQP.SMF.SYS*.**

Job to run the accounting report:

```
//ACCOUNT  JOB (TYYYYY,XXXX),
//         CLASS=6,MSGCLASS=P,MSGLEVEL=(1,1),REGION=0M,
//         NOTIFY=DB00SXA,
//  TIME=(50)
/*JOBPARM S=SYSC
//*************************************************************
```

```
//AUTH     EXEC PGM=ACF2NCB
//*************************************************************
//*************************************************************
//*************************************************************
//STEP1    EXEC PGM=FPECMAIN
//STEPLIB  DD DSN=SQP.OMEGXE52.SMPE.TKANMOD,DISP=SHR
//INPUTDD  DD DSN=SQP.SMF.SYSE.G2660V00,DISP=SHR
//SYSOUT   DD SYSOUT=*
//DPMLOG   DD SYSOUT=*
//JOBSUMDD DD SYSOUT=*
//ACRPTDD  DD SYSOUT=*
//SYSIN    DD *
GLOBAL
FROM(06/01/15,00:01) TO(06/01/15,23:59)
 INCLUDE(SUBSYSTEMID(DBE1))
ACCOUNTING
REPORT
LAYOUT(SHORT)
EXEC
```

Please find the sample accounting report below for DBE1 subsystems.

DBE1 accounting report.txt

As per this above report we could find two AQT queries ran by the users. Find the details below,

```
1)JOSEPT                                   4       4    0.00    1.00           1.00
40.206210     2.50     0.00
  aqtv9_x6                                4    0.00    0.00    0.00      41.109528
5.944577    38.25        0
                                          0    0.00    0.00    0.00       5.944803
872.7K 27474.50

     -----------------------------------------------------------------------------------
-----------------------------
     |PROGRAM NAME        TYPE      #OCCURS  #ALLOCS  SQLSTMT  CL7 ELAP.TIME   CL7 CPU
TIME  CL8 SUSP.TIME  CL8 SUSP|
     |SYSSH200            PACKAGE         4        4     2.50      40.206066
5.944481      28.571494     21.1K|
     -----------------------------------------------------------------------------------
-----------------------------

     -----------------------------------------------------------------------------------
     |REQUESTER         METH  #DDFS    TRANS  #ROLLBK  #COMMIT  SQLRECV  ROWSENT  CONVI|
     |::159.108.159.44  DRDA      4      N/A        0        4     2.50    51.00   0.00|
     -----------------------------------------------------------------------------------
2) PA                                      27      27    0.00    0.89           1.00
0.016038     0.56     0.00
  aqtv9_x6                               27    0.00    0.00    0.00       0.454371
0.004417     1.00        0
                                          0   12.93    0.00    0.00       0.004805
93.19     3.63

     -----------------------------------------------------------------------------------
-----------------------------
     |PROGRAM NAME        TYPE      #OCCURS  #ALLOCS  SQLSTMT  CL7 ELAP.TIME   CL7 CPU
TIME  CL8 SUSP.TIME  CL8 SUSP|
     |SYSSH100            PACKAGE        27       27    15.63       0.015456
0.004375       0.005057      0.93|
     -----------------------------------------------------------------------------------
-----------------------------

     -----------------------------------------------------------------------------------
     |REQUESTER         METH  #DDFS    TRANS  #ROLLBK  #COMMIT  SQLRECV  ROWSENT  CONVI|
     |::159.108.153.39  DRDA     27      N/A        0       27    15.63  2722.52   0.00|
     -----------------------------------------------------------------------------------
```

Similarly, we can find the high CPU utilized threads info from the accounting report, using the ELAP time.

4. Use of History tables and Routine Maintenance activities on DB2 Objects

The infrastructure team suggested the usage of a history table and archiving the oldest rows from the DB2 application tables which are mostly used in the CICS Online –DB2 Programs.

<table>
<tr><th><b>Parameter</b></th><th>Use of History Tables</th></tr>
<tr><td><b>Observation</b></td><td>• Some of the Application tables are very huge in size and they contain millions of rows.
<table>
<tr><th>Table Name</th><th>Total Number of Rows</th></tr>
<tr><td>KIRMOUSG</td><td>166,215,805</td></tr>
<tr><td>KICHRG</td><td>108,045,567</td></tr>
<tr><td>KICSL</td><td>100,409,002</td></tr>
<tr><td>KIFEVDTL</td><td>73,953,608</td></tr>
<tr><td>KICADL</td><td>66,849,379</td></tr>
</table></td></tr>
<tr><td><b>Recommendation</b></td><td>• Use History tables to store the data which may not be used for a periodic reporting
• This will ensure the base tables have limited number of rows which can result in better I/O and hence better performance
<b><i>This purely depends on how the data can moved as per Application's Requirement</i></b></td></tr>
</table>

Although REORG jobs and RUNSTATS jobs were already functioning in the system, all of the maintenance jobs were reviewed and corrected. The critical financial tables which were widely being used in the times of crisis were studied with extra care and nothing was missed in maintenance. This helped a bit on faster processing too.

A Brief approach of Maintenance activity is listed below:

Parameter	Routine Maintenance activities
Recommendation	• Identify most critical tables related to financial close and monthly reporting • Ensure the monthly REORG activities are diligently followed and all critical tables are included in the activity • Removal of unused temporary tables from critical table spaces • Removal of obsolete database objects from the Sub-system

5. Changes to Indexes in the huge tables

Large financial tables were studied for new indexes, implemented, and the relevant packages were rebound. This helped in improving queries which had date ranges queried against the large financial tables. Some of the expensive queries were first studied and underlying table-spaces were investigated for additional indexes.

6. Monitor for Deadlocks in DB2:

-904 SQL Code in DB2, which is for deadlock, was monitored regressively and the application team was informed if there was a deadlock situation arising in the production system. A possibility of automating the -904 reports to the application team was devised and in development.

Note: We followed mainframe system recommendations which are not listed here, butwhich helped in easing the critical issues.

Conclusion:

With the help of these steps and overall approach, the problem was contained. We started to show best results starting at 3 weeks and most of the critical P1 and P2 tickets were drastically reduced to 50% within 3 months.

The Capacity and performance management was not done in a radical way to change the application or add processor /additional CPU to the infrastructure, but followed several effective ways in a gradual phase –both from the database side and the system side as described above.

Adding additional CPU capacity was in plan but not in the near term. Without adding additional CPU capacity, a consistent amount of the problems were arrested in the mainframe environment. These were some of the best practices and proactive approaches that were followed. Many of these approaches can be tailored according to their local mainframe shop, but some of them would be applicable to almost all mainframe environments.

The cost effectiveness of not bringing in a radical approach to handle the problem was an upside in the whole course of these activities. With less cost and as an on-going activity, we could able to turn around the critical problems to business as usual. There was no need to shift the workload to a different background at any given point in time.

References:

- *DB2 10 for z/OS Performance Topics- SG24-7942-00*
- *Monitoring Performance from the IBM Tivoli OMEGAMON Enhanced 3270 User Interface- SH12-7056-00*
- *System Programmer’s Guide to: Workload Manager - SG246472*
- *IBM Tivoli Workload Scheduler: User’s Guide and Reference - SC32-1274-15*

Re-inventing ICT but Not the Wheel

Flavio Gaj - Silvio Antonio Varagnolo
PriceWaterhouseCoopers Italy

The complexity of ICT (Information and Communication Technology) and their role in supporting businesses are out of discussion. Digitalization will be the challenge in the next coming years. Greater emphasis on quality as perceived by customers, cost control, and a more rigorous and quantitative way of measuring and managing services will become imperative in the next few years. It will be the key to success for the smarter ICT Company. The approach proposed inverts views and priorities: the constraints and rigidity of ICT no longer condition business, business priorities direct and guide the ICT operations. It retrieves and adapts to ICT well-established manufacturing criteria and concepts such as the bill of materials and others.

Introduction

"ICT Service Management" (Information and Communication Technology Service Management) is not a new term. It has been increasingly talked and written about for several years now. Reference models, guidelines and certification systems abound. Of these, the Information Technology Infrastructure Library (ITIL) is the most consolidated, exhaustive and closest to the problems of ICT Service Management.

ITIL was developed by the UK in the 1980s in response to the growing dependence on ICT. This framework remained largely ignored and little used until the mid-1990s when the Internet era began and the consequent opening outwards of corporate information systems made the management methods used until then both obsolete and outdated. ITIL exploded and it quickly spread throughout the world, USA included. Subsequently it gave rise to the issuing of support standards including ISO/IEC 20000 (previously BS 15000) which is inspired by many elements of ITIL for ICT Service Management.

Information and Communication Technology (ICT) has been considered a technical, or engineering, discipline for some years now, but it has not always been so. For a long time after its beginnings in the 1960s, it was not considered a specific discipline that merited consideration, academic or otherwise, but solely a combination of electromechanical machines designed for a specific purpose (i.e. managing information).

Background

The first computing centers were housed in the historic centers of cities where, traditionally, the banks were located, as they were among the biggest users, just as they are today. Computing centers were also appendages of industrial plants and were simply considered ancillary instruments for the administrative or productive sectors. The ICT sector was called Data Processing which depicted exactly what it was expected to do: calculate more quickly and, if possible, with fewer errors what had previously been done manually.

The computing centers were also called "electric accounting centers" and although in the mid-70s remote processing (Tele Processing as it was called) began to establish itself, the concept of technology and services based on it were still far off. Internet existed only in the boldest imagination that speculated about how processing power and storage could one day become commodities that, like electricity, would be available through outlets, regardless of the physical location of the technical infrastructures (this was the cloud ahead of its time…).Users were few and only some – and of these, just a small number– relied on ICT to perform their work.

Moreover, users worked during the day so the machines and computing centers were available for maintenance, tests, and the installation of new devices etc. only for a few hours during the night when the batch and maintenance procedures had ended. If the available night hours were not enough, weekends were used.

The technology was electronic and partly mechanical while the processing of information was simply an unstructured outcome, which was not explicitly recognized but, more especially, not studied. Software was an ephemeral concept; no one dreamt of paying for it, it was an appendage of the machine.

This evolution was accompanied by a slow and gradual recognition in the 1960s of the need for high-level professionals who specialized in the processing of information and its implications, from databases to languages, from networks to protocols, et cetera.

Computer Science was thus established and still today it produces technological innovation and tools of various kinds. In fact, the first faculty of Computer Science in Italy dates back to the late 60s.

Similarly, today, the need to have service experts working alongside ICT technical experts is felt more and more - even after the explosion in the number of ICT users and the opening of company information systems to the outside world thanks to the internet. Service experts who can design and implement end-to-end delivery systems and make the user experience positive in all its phases from design to testing, operation to support, who facilitate and structure communication with users in a shared, effective and clear way.

The need to identify and introduce Service experts is also strongly felt in other sectors due to the constant growth - at least in the Western world - of the service sector and of the decreasing importance of product in favour of the service built around the product. The acronym SSME (Service Science, Management and Engineering) proposed by IBM has become popular in the last few years. The SSME approach can be defined as "*the application of scientific, management and engineering disciplines to tasks that an organization performs for the benefit or advantage of another (namely, services). SSME aims to make improvements in productivity, quality, performance, compliance, growth and learning more predictable in work-sharing and risk (co-production) relationships. SSME is the study of service systems with the aim of improving them*".

SSME as a concept was announced in 2006. Many companies have embraced it and actively support universities around the world in contributing to the development of the so-called ***T-shaped professional***. This is a professional figure that, while having an in-depth knowledge of a single discipline, has sufficient knowledge of others relating to it to give them a systemic vision of the organization of service delivery which takes into account, in a coherent and coordinated way, technical, organizational and financial aspects as well as people management.

It is interesting to note that SSME aims to create models and awareness that in manufacturing industries that gave rise – following the efforts of the first half of the 20th century – to definitions, taxonomies and models known as Material Requirement Planning (MRP), MRP II, Enterprise Resource Planning (ERP), et cetera, which resulted in supply chain management methodologies which are now part of the common heritage and taught all over the world.

ICT function mission in the Digital and Cloud era

Sometimes we imagine an ICT function mainly as a software firm. This is only partially true, because the business more and more relies on standard software packages, or on custom software developed on order by third parties. What end users really see is not software, but services, able to be up-and-running in the agreed ways, quality and time.

In spite of this, often the management of the delivery of the ICT services does not have the rank of a high-level organizational unit, autonomous in its own right and able to act as a peer counterpart of the suppliers of components (Applications and Infrastructure). Usually ICT operation is a low-level organizational unit inside Infrastructure division, with difficult and troubled relationships with application development divisions and infrastructure division.

A first superficial analysis may lead you to think that the ICT of the company – in the current time evolving to cloud and outsourcing - is simplified because it is delegated to external suppliers that, supposedly, know the work better than a company ICT department (true in many cases).
In reality, the complexity does not diminish but is shifted onto another level. Alongside the diminished need of technical skills for product use (operating systems and middleware), other skills such as architectures, sizing, service integration, problem solving and customer management come to the fore and become the critical factor for success.

The ICT department focus needs to shift from software products design and implementation details (handed over to product and service providers) to an architectural vision that ensures coexistence and the proper functioning of a system composed of heterogeneous macro-components.

Service management acquires even greater importance. It is unthinkable, in fact, to leave to the end user, the burden and responsibility - for malfunctions and small maintenance - of interacting with the Service Desks of the various suppliers, though useful and necessary. The ICT department within the company will have to organize its own Service Desk that is able, on one side, to deal with its own customers and, on the other, with the multiplicity of suppliers described. Anyone with experience of managing service providers can understand that this does not mean only intercepting and sorting out communications, but also and above all, using knowledge and diagnostic skills applied to a global architecture that only the ICT department masters, the knowledge of single providers here can only be partial and limited.

Figure 1 shows the complexity of a real ICT architecture, as an outcome of many years of addition of technology, application, infrastructure, projects and suppliers.

The ICT function needs to manage numerous entities/units. There is custom software (developed in-house) to be maintained which requires technical skill and programming. Relationships with suppliers of application packages must be managed, each of which can have different support, maintenance and development policies.

Just think about the adjustment of applications as the operating system and middleware evolve; it is not uncommon that different vendors have different adjustment times for their applications as infrastructures evolve thus indirectly imposing the need for duplicate platforms to coexist, albeit for limited periods.

The same considerations apply to the various applications managed in Cloud aggravated by the fact that where the application providers and Cloud platform providers are different it is necessary to coordinate their work. This is especially true for Service Management activities, which, by their nature, are not easily plannable and predominantly reactive – when things go wrong, urgent reaction with maximum possible efficacy is required.

Problem determination in these cases is very complex. Often the root of a disruption is in the communication between the two components, or in the interfaces between them, with responsibility that is not immediately visible or identifiable. If the collaboration of two or more providers is necessary to get a diagnosis then authoritative and credible coordination is essential. Nor should it be forgotten that in the event of disruption for the end user, the only concern of each provider will be to prove that they are not responsible, which is a very different thing from trying to solve the problem efficiently and in the shortest possible time.

Gearing up ICT structure for the Digitalization challenge

The central point emerging from these reflections is the evidence of the need for a high-level function whose scope is service. This is not a unit of lower hierarchical level, lost in technology management and relegated solely to low added-value operating activities but a real ***ICT Services Management*** of equal dignity to *Infrastructure Management* and *Application Development Management* that is in charge of the development and operational management of a new industrial model based on a relevant Information Systems.

The word “MANAGEMENT” should be underlined in the function responsible for designing the Service Catalogue (in collaboration with the Customers), defining classification parameters, organizing and managing the description of services in terms of components and related providers, obtaining the necessary indices from the providers, managing overall costs control. The diagram in Fig. 2 can exemplify the main organizational functions of governance and control.

An outlandish point is the fact that just designers of complex Information systems think that a complex structure such as an ICT function could be managed without the support of an adequate Information system….How many ICT structures can rely on an integrated database allowing a holistic, analytic and comprehensive vision of its own entities? An example of a logical data model (illustrative) is shown in Fig. 3.

The picture in Figure 4 shows a functional illustration of an ICT structure. The picture is simplified; it focuses in particular on the functions and flows related to the service. The main figures authorized to interact with ICT are shown at the top: Users (Final users using the Service) and the Customers that guide the design, development and maintenance of the Information Services through the ICT Demand Management function. Figure 4 shows the ICT structure with its fundamental functional blocks: ***Service Management, Infrastructure Development, Application Development, Relationship management.***

How to face the Change

Establishing a new operating model is fact very difficult for several reasons.

The maturity of ICT management is the first obstacle as it is still poor due to a short history and rapid growth. Reconciling ICT technical knowledge and managerial skills is very difficult because we tend to learn from the environment in which we live and the ICT environment is much unstructured and essentially artisan. For this reason, even the brightest young people tend to develop a distorted view (they go from being a nerd to being a manager, or – worse – they become a manager without a deep knowledge of the ICT profession) and perpetuate the status quo. This explains why there are many cases where companies appoint people who come from a business or production experience as CIOs; it is to inject skills and managerial perspectives into a community that failed to develop them independently. It is a difficult challenge; for a single person to change a consolidated organization with a strong culture is frequently impossible. The organization changes him as often happens ….

Perhaps we must seize opportunities of strong discontinuity such as an outsourcing initiative, or a merger, to question everything and start over.

Another point worth attention is the awareness of the need for an ICT support system that clearly requires the simultaneous definition of a series of organizational and terminology bases. No one – today – would dream of establishing any business without a proper information support system, why do our communities think they can do without one …?

An experienced reader will certainly have noticed that the proposed model is also valid where all components (applications and infrastructure) are outsourced, and this can become a formidable instrument of governance and control.

Changing ICT people - their awareness and their professional profile - will come not through trite stereotypes about change, but through a reinterpretation of how ICT should be managed in industrial and service terms.

Training initiatives for the transformation of a structure should therefore be set up on a few fundamental steps:

a. **The definition of an operating model**
   This proposal relates to the management of the Service. A similar and consistent model (complete with terminology and glossary) should be prepared for the development of information system projects and for their maintenance. Note carefully! Not on software development *only*; even if software development is outsourced (or is based on packages) Information System projects exist! And must be managed ....

b. **Spreading the model in the organizational structure**
   This is a critical step and is the real driver of change. It should be the heart of training and change and should also serve to gather and manage objections and suggestions for change, both to involve operations people and make them feel part of the journey, and to, objectively, identify details that can have escaped in the design of the new operational model.

c. **Testing the model in a pilot environment (prototype)**
   It is advisable to start soft and by degrees, in spite of having very clear ideas in general terms and on where you want to go. Being concrete does not mean solving problems as they arise, but being determined to reach an ambitious final goal through small coordinated, integrated and consequential steps.

d. **Operational diffusion of the model**
   The previous phase will have served to refine the model which in this phase will be spread to environments other than the pilot.

All this could be accompanied by traditional *soft* initiatives on change but they will complement it but will not be the fundamental part. That is the one described above. Do not get your hopes up that shortcuts can be taken; such radical changes are fraught with obstacles and are time-consuming.

Conclusions

ICT profession today is in trouble; it should now be clear that the change needed is large and that only a conscious and sufficiently long-term approach can be successful. Management must be braver and more daring in making decisions: a short-term vision cannot support a conscious and mature change.

People's viewpoint needs changing, maybe a new professionalism needs to be created (Service specialists) that will work alongside the classical ICT professional. Service science could also help our community to grow and evolve. But the author is convinced that it will be appropriate to move quickly, to ride the push for change and prevent our profession and our skills from migrating to countries that are more sensitive and better able to seize this opportunity.

The concepts summarized above have been deepened in a book that can be found at the following links:

Paper form:

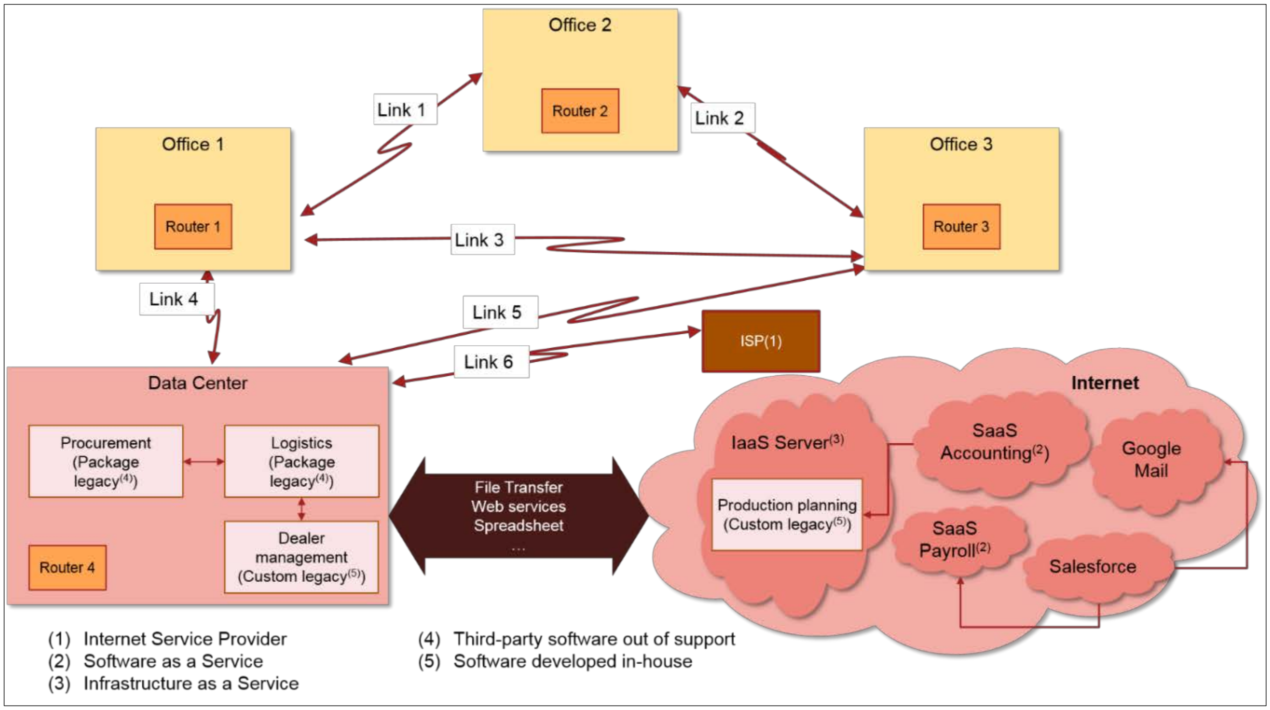
http://www.shopping24.ilsole24ore.com/sh4/catalog/products/libri.jsp?productId=prod2400018&categoryId=SH246056&locale=&productPage=true

Kindle form:

https://www.amazon.it/s/ref=nb_sb_noss_1?__mk_it_IT=%C3%85M%C3%85%C5%BD%C3%95%C3%91&url=search-alias%3Daps&field-keywords=flavio+gaj

From the Italian PWC site:

http://www.pwc.com/it/it/publications/reinventing-ict.html



(1) Internet Service Provider
(2) Software as a Service
(3) Infrastructure as a Service
(4) Third-party software out of support
(5) Software developed in-house

Fig. 1 – Example of a real ICT architecture

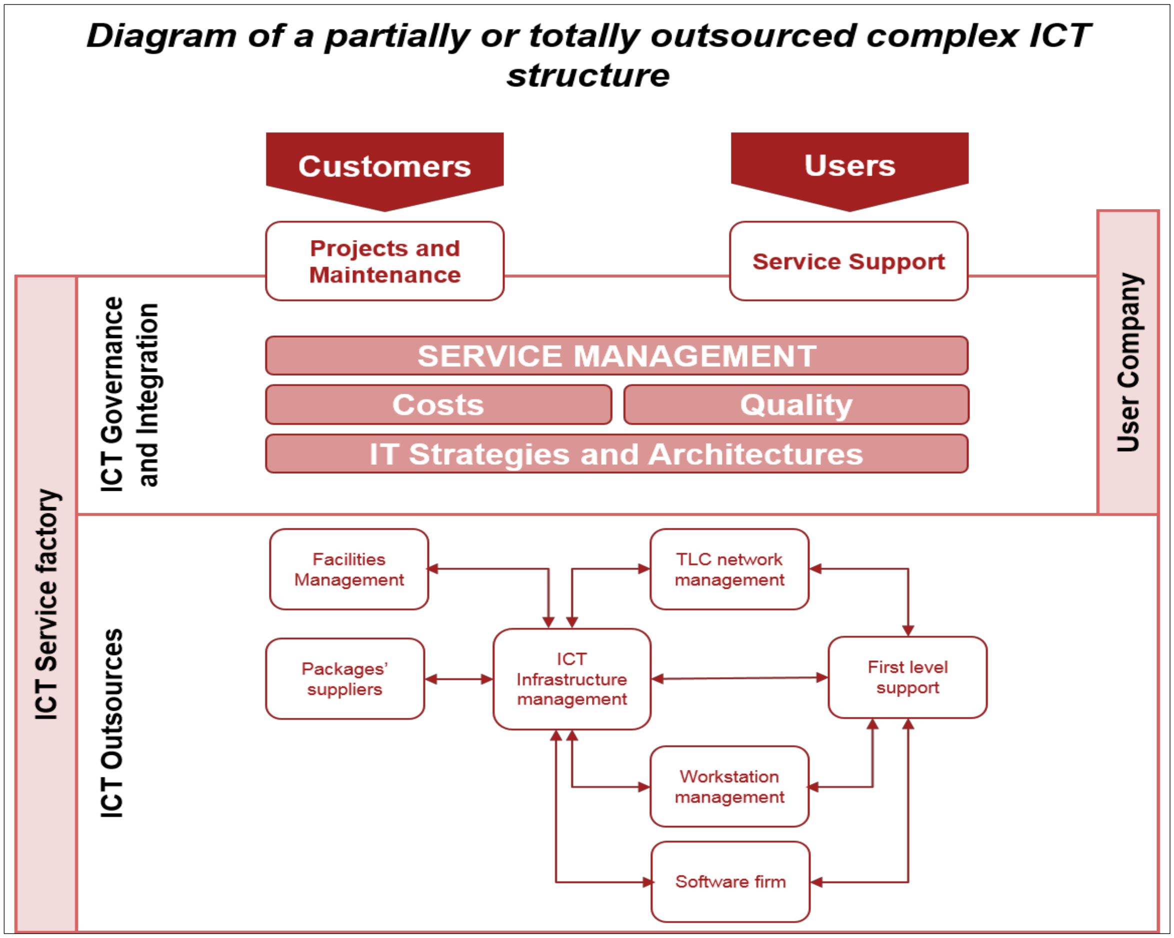


Fig. 2 – Diagram of a complex ICT organizational structure

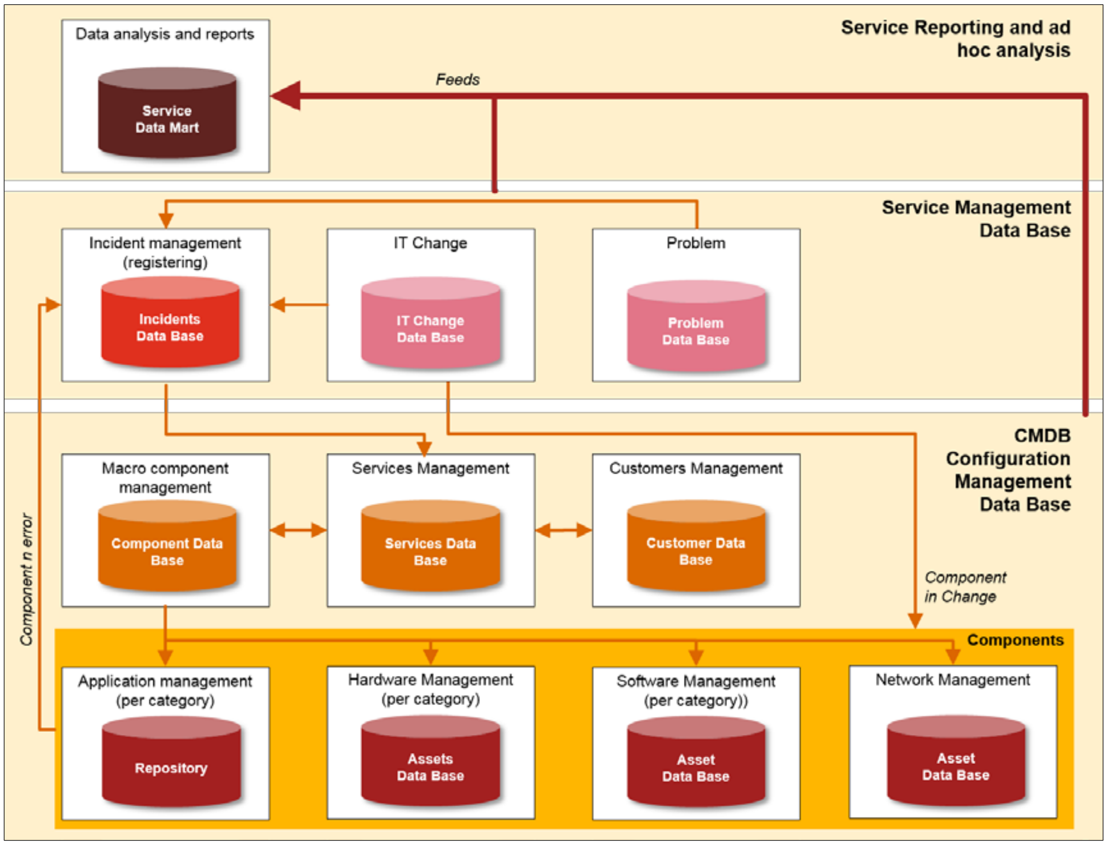


Fig. 3 – An illustrative example of the Data Model of an ICT Service factory

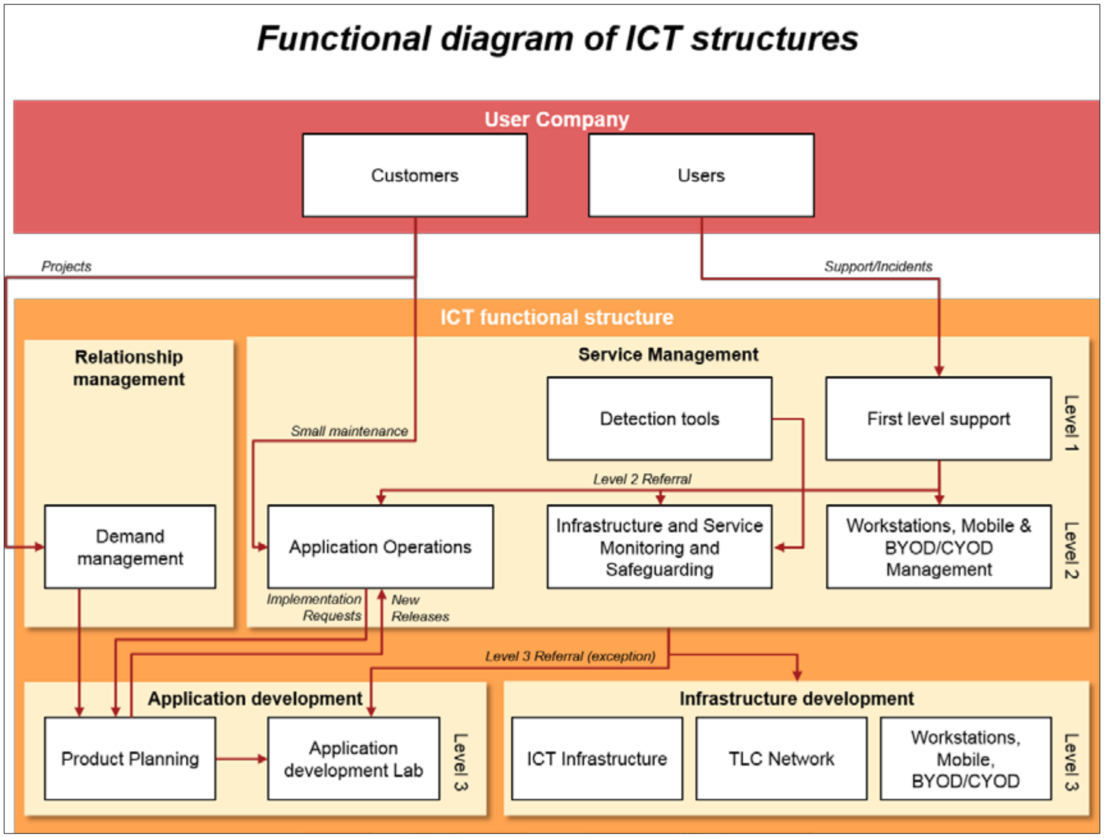


Fig. 4 Functional diagram of an ICT structure

CMG
Computer Measurement Group

Computer Measurement Group
3501 Route 42, Suite 130, #121, Turnersville, NJ 08012
856-401-1700 | cmghq@cmg.org