

# INFORMATION SYSTEM AND PERFORMANCE MEASUREMENT LIFE CYCLE

**Sandip C. Patel**  
**Suneel K. Maheshwari**

## **I**NTRODUCTION

In today's technologically challenged competitive environment an effective information system is critical to the success of a business. The performance of such information system is vital to its success. The performance of an implemented information system is many times the single most important aspect of an information system for a business. However, the measurement of an information systems' performance is elusive and is sometimes overlooked during the actual system development cycle.

SDLC has following phases – planning, analysis, design, implementation, support and maintenance.<sup>1</sup> The measurement of the systems' performance is restricted to the implementation phase of the traditional system development life cycle (SDLC). The measurement tools are designed to measure the performance of applications or the subsystems only. These tools are thus burdened with the task of evaluating the entire systems' performance for which they are not equipped. To make matters worse the rigidity of the process does not allow the systems programmer/analyst to easily incorporate the changes suggested by the outcome of performance measurement.

The problem is that the cost to incorporate changes suggested by the outcomes of the performance measurements increases as the SDLC advances. This is because the stages of SDLC will have to be repeated to implement the desired changes. Boehm (1981) supports the contention of adverse financial implications of such changes at later stages in the SDLC.

We examine SDLC and focus on its performance measurement phase. The paper evaluates various performance measurement tools and their role in the implementation phase of the SDLC. We recognize the importance of performance of the information system and suggest a performance cycle parallel to the SDLC starting with the planning phase. Specifically, the paper suggests the introduction of a Performance Measurement Life Cycle (PMLC) which parallels the SDLC. The new PMLC introduces the performance measurement tools earlier than the SDLC does. This would lead to not only the effective use of measurement tools but also will facilitate the process of implementing changes suggested by the performance tools.

In the following sections we look at the tools used for the performance measurement in the SDLC. We have categorized these performance measurement tools into two groups: system performance benchmarks and the system performance products. The study of tools is followed by the description of the proposed PMLC and conclusions.

---

*1 Planning involves identifying business problems and planning the goals to be achieved by the information system. System analysis involves analyzing business processes, problems, and suggesting solutions. In the design stage technical details for the solution proposed in the analysis phase is developed. System implementation puts the developed solutions into operation followed by the system support and maintenance stage.*

---

## Performance Measurement Tools

### System Performance Benchmarks

Benchmarks are designed to measure the performance of a specific task or a subsystem such as memory management. Benchmarks are often used to measure general things like graphics, I/O [input and output], compute (integer and floating point), performance, etc. But most measure more specific task like rendering polygons, reading and writing files, or performing operations on matrixes (Sill, 1993-95). Some of the benchmarks are:

#### (1) Intel® Performance Evaluation and Analysis Kit (IPEAK)

Intel's IPEAK is a family of platform performance and integration tools. Currently, IPEAK is comprised of seven tools (listed below) for optimizing various components of the PC platform. For an OEM, Independent Hardware/Software Vendor, the IPEAK tools will help optimize the design of products for the PC platform.

- a) The Intel WDM I/O Subsystem Performance Monitor (IOMon) is a software tool that enables the tester to verify both the functionality and performance of their hardware devices and device drivers.
- b) The Graphics Performance Tool provides an integrated graphic user interface (GUI) environment for analyzing graphics hardware and software performance on the Intel Architecture platform. It includes hardware performance analysis, software performance analysis, workload/scene analysis, and API analysis.
- c) The IPEAK Baseline AGP System Evaluation Suite (IBASES), provides a collection of tools that evaluate AGP performance for optimum hardware and software integration.
- d) The Intel Power Management Analysis Tool (IPMAT) helps in the evaluation and qualification of systems that support the Advanced Configuration and Power Interface (ACPI). IPMAT can be used for vendor selection and qualification by checking for, and exercising the ACPI power management support of hardware, devices, and drivers.
- e) Storage Tool-kit is designed to aid in the performance improvement and selection of storage devices. It includes the capability to manipulate and analyze system-level disk I/O traces, rank drive performance, and perform low-level drive performance analysis.
- f) DQUICK is a software tool that aids in the building of systems, which include DVD, specifically host-based DVD playback. This tool looks at all components, which interact for DVD playback, such as audio, graphics and video and reports back on whether the platform is optimized.
- g) The 1394 Tool-kit is a software suite that helps monitor performance and verify the operational stability of 1394 PC drivers, system bus and peripherals.

#### (2) FreeBSD Inc.:

FreeBSD is an advanced BSD UNIX® operating system for "PC-compatible" computers, developed and maintained by a large team of individuals. Some of the benchmarks maintained by FreeBSD are listed below:

- (a) bonnie-1.0: Performance Test of File system I/O
  - (b) bytebench-3.1: The BYTE magazine benchmark suite
  - (c) dbs-1.1.5 : A distributed network benchmarking system
-

- (d) hint.serial-98.06.12: A scaleable benchmark for testing CPU and memory performance
  - (e) iozone-3.9: Performance Test of Sequential File I/O
  - (f) lmbench-1.1: A system performance measurement tool
  - (g) netperf-2.1.3: Network performance benchmarking package
  - (h) netpipe-2.3: A self-scaling network benchmark
  - (i) nttcp-1.4: A client/server program for testing network performance
  - (j) postmark-1.11: NetApps file system benchmark
  - (k) rawio-1.0: Test performance of low-level storage devices
  - (l) tcplblast-1.0: Measures the throughput of a tcp connection
  - (m) xengine-1.0.1: Reciprocating engine for X
- (3) Other benchmarks are available from companies such as PC Magazine, AIM Technology, BYTE Magazine, Computerworld, Datamation, PC Week, and DataPro.

The benchmarks measure specific aspect of a computer system but they do not provide an indication of the application performance such as an application producing drill down reports for a management information system. A set of benchmarks is not feasible to measure all systems performance because of the diverse nature of the applications run onto a system. "One system may be excellent at performing simple update-intensive transactions for an online database; but it may have poor performance on complex queries to that database. Conversely, a system that excels at decision-support queries may not even allow online transactional access to that same data".<sup>2</sup> To solve this problem, domain specific benchmarks were evolved. Domain specific benchmarks specify workloads for a typical application on that problem domain. Some of such benchmarks are:

- a) TPC BM™ A: Online Transaction Processing including a LAN or WAN network.
- b) Wisconsin: Relational Queries
- c) AS3AP: Mixed Workload of Transactions, Relational Queries, and Utility Functions.

However, the domain specific benchmarks fall short of measuring the system performance since a system performs multiple cross domain applications.

### **System Performance Products**

The system performance products measure the performance of various components of the information system.

#### (1) Candle Products:

Candle Corporation provides several products for measuring application performance. Among them are:

##### (a) Measuring Application Response Time:

- 1. eBA\*ServiceMonitor™ — Measures response times and usage from the customer perspective

---

<sup>2</sup> The Benchmark Handbook © Copyright 1991-1998 by Morgan Kaufmann Publishers, Inc.

2. eBA\*ServiceNetwork™ — Analyzes customer data over time and publishes executive-level reports
  3. Candle Response Time Network (RTN™) — Web-based subscription service that delivers response-time analyses — how, when and where they're needed. RTN tracks computer response time for each transaction, time spent waiting for the user to make another request. It can analyze the types of actions the operator performs on the screen.
  4. ETEWatch™ precisely measures end-to-end application response time from the users' point of view
- (b) Candle Command Center: It is a systems management tool for optimizing an organization's computing resources and maximizing business application availability.
- (c) Omegamon™ II performance monitor is used for real-time and historical analysis of performance.

(2) Precise Software:

Precise Software Solution's Precise/Pulse provides performance monitor for Oracle database applications.

(3) HINT™:

HINT was developed by Gustafson and Snell at Ames Laboratory in Iowa. HINT works on any architecture and measures the capability of a digital computer in graphs.

(4) Tempus:

Tempus helps predict the performance of a real-time system early in its development by examining the source code and makes the scientifically-based measurements that correlate with the software's real-time performance.

Despite wide use of the traditional approach, it has flaws such as that of being rigid. Once the analysis phase is complete, any change is difficult and expensive to incorporate into. The later the problem is detected, the more expensive it gets. Barry Boehm (1981) has done the detailed work on the financial aspects of implementing the change.

## **Performance Measurement Phase**

In the traditional SDLC a typical system performance analysis starts toward the end of the system development cycle which is to deliver the system into operation. Under the System acceptance/validation testing, system performance is measured for its adequacy. The system acceptance test is the final opportunity for end-users, management, and information systems operations management to accept or reject the system (Whitten and Bentley, 1998). Measuring a system is, in effect, the final task of systems development (Stair and Reynolds, 1999). Thus, if a change has to be made to accommodate a performance upgrade, it would be very difficult and expensive to implement.

### **Suggested PMLC**

Performance is such an important aspect of the system that performance measurement should have its own separate cycle. The Performance Measurement Life Cycle (PMLC) should run in

---

parallel with the SDLC cycle. The PMLC must identify areas of the performance measurement for each layer. We suggest the following four-layered model for the performance measurement:

**Layer 1: Business Layer:**

Business performance measures should be defined and specified in this layer. For example: the speed at which the transaction processing system needs to generate its output. Business layer may correspond to the planning and systems analysis phase of the SDLC.

**Layer 2: Application Layer:**

In this layer the analysts need to determine the performance indicators and their criteria for acceptance or rejection of the system. Examples of indicators in the application layer include time taken by each business transaction, system downtime, transactions completed per second.

At this stage it needs to be decided what application performance measurement tools should be used and whether to purchase the existing tools or develop new tools. This stage may correspond to systems analysis and design phase.

**Layer 3: Technology Layer**

Performance measures are further refined during the application layer stage. Technical parameters of the performance measures are specified at this level. The criteria for acceptable hardware performance and software performance such as computer memory or input/output performance should be decided in this layer. Benchmark criteria for the acceptance of the results are specified.

**Layer 4: Execution Layer**

Execution layer starts with the design phase of the SDLC and provides continuous feedback to the SDLC. Although, the design might not be complete, execution can take place on planned hardware and some prototypes of the applications. This phase will continue through the system implementation phase of the SDLC.

Once the above top down analysis is completed, the feedback provided should be implemented starting from the technology layer to the business layer, in the reverse order. Such an implementation would ensure that the technical implementations would meet the business objectives defined during the first layer.

**Conclusion**

Our paper highlights the two major problems with the traditional SDLC. These two problems root in the rigid nature of the SDLC and postponement of the performance measurement to the implementation phase of the SDLC. The rigidity of the process does not allow the systems' programmer/analyst to easily incorporate the changes suggested by the outcome of performance measurement. Such changes are difficult and expensive to incorporate at the implementation phase of the SDLC.

We suggest a multi-layered performance measurement life cycle where development of the broad performance measurement parameters starts with the initial phase of the traditional SDLC. These performance measurements are refined as we proceed along the subsequent layers of the PMLC.

The new PMLC will give the performance objective the front seat in the development cycle. We

---

realize the inclination to start measuring the system performance only after it is implemented. But the performance is such a crucial aspect for the success of the information system that, in our opinion, it deserves a separate but closely associated cycle to the SDLC. Also, the performance measurement tools will have the right place within the PMLC.

## **References**

Boehm, Barry W. (1981). *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice-Hall.

Stair, Ralph M. and George W. Reynolds (1999). *Principles of Information Systems: A Managerial Approach*, Course Technology ITP.

Whitten, Jeffrey L. and Lonnie D. Bentley (1998). *Systems Analysis and Design Methods*, Irwin McGraw-Hill.

---