Presented at CMG 2006, December 3-8, 2006

Full Lifecycle Performance Engineering

Amy Spellmann
HyPerformix, Inc.
amy@hyperformix.com

Richard L. Gimarc
HyPerformix, Inc.
rgimarc@hyperformix

Christopher Lee
Wells Fargo Bank
leecw@wellsfargo.com

In this paper we describe and demonstrate Performance Engineering methods and techniques that apply across the software development lifecycle. This methodology, which is based on reusable tools and repeatable processes, can be applied at any stage of the lifecycle. The paper includes a case study of a retail banking application where Performance Engineering was successfully applied during design, test and production.

1 Introduction

In the quest for good application performance, IT professionals tend to use Performance Engineering (PE) methods and techniques in an inconsistent and intermittent fashion at various stages in the software development lifecycle. Only when there is a serious performance problem are these proven methods applied. Why do we wait until there's a problem?

Everyone has heard the typical answers and excuses:

- *It's too difficult* – The techniques are beyond the capabilities of our staff and we can't afford to hire high-priced consultants.
- *It takes too long* – We have trouble meeting our current deadlines, adding extra tasks is only going to slow us down and delay delivery.
- *The timing is never right: it's either too early or it's too late* – Too early usually means that the application is still in the design stage and we don't have any "data" or measurements to characterize performance. Too late often means the application has already been deployed and we are seeing unexpected scalability problems.

As performance practitioners we need to ask ourselves:

- How can we apply proven PE methods and techniques to deliver continuous value that ensures application performance?
- How can applications be managed effectively and efficiently, from a performance perspective, to support the needs of the business?

Application performance is paramount; too many stories have been told about lost revenue and opportunities due to poor performance. Over the past 15-20 years, a rich collection of PE best practices have been developed and refined. These practices have been successfully applied at discrete points across the application lifecycle; design, test, and production. The missing ingredient is a methodology that ties these independent actions together so that their combined value and benefit can be leveraged for the greater good.

Instead of viewing PE as a set of "best practices", we need to take a more holistic approach. For example, consider the following:

- How can we leverage, reuse, and extend the PE work done in the design stage when the application moves into test?
- What Work Products do we produce in the test stage that can assist with our continued monitoring and management of the application as it moves into production?
- Why don't we view the tools, knowledge, and processes produced by PE efforts as intellectual capital that is deemed critical to the future success of the business?

This paper describes a practical, efficient, and proven delivery mechanism to apply Performance Engineering via a Roadmap with reusable Work Products. First, we describe the PE Practice and the Roadmap. Next, the reusable Work Products are defined. The paper concludes by illustrating the process with a case study from a retail banking application.

2 The PE Practice

Performance Engineering is defined as follows:

> *The application of engineering disciplines to institutionalize performance practices throughout the application development lifecycle.*

The PE Practice is the consistent application of PE methods and techniques "*to provide a level of assurance that applications, systems, and services provided by IT satisfy the performance requirements of the business owners*" [SPEL2000]. The main point to note here is that the Practice provides service to the business; they are the business-critical resource to mitigate risk.

The ideal starting point for a PE effort is in the design stage of a new application. The requirements have been stated and the architecture group is starting their creation of a system and application design that meets the application's functional and performance requirements. Much has been written about the application of PE during the development of new applications (e.g., see [SMIT1990]). Methodologies and techniques have been described that assist the architects in their creation of a design that meets the application's performance requirements.

Although the design stage is the preferred starting point, the PE Practice can provide value by active involvement in all lifecycle stages. In [SPEL2000] the authors describe the concept of defensible deliverables, where performance is "certified" as the application moves from stage to stage. Primarily, PE efforts focus on three key lifecycle stages:

- **Design** – PE services include early predictive studies and workload analysis to evaluate the feasibility of the design, develop performance budgets [ZAHA1993, ZAHA1995], develop an initial estimate of infrastructure requirements, and identify sensitive application components.
- **Test** – PE services are used to extend the scope of load testing, sharpen the focus of performance testing and contribute to the deployment plan [GIMA2004].
- **Production** – PE services are used for ongoing capacity planning and problem diagnosis [LETN2005].

The success of a PE Practice is dependent on the ability of its practitioners to be flexible (apply PE best practices at any lifecycle stage) and efficient (leverage and reuse PE Work Products). A keen understanding of the Work Products created and applied at each stage will enable the Practice to rapidly customize their service and deliver results in a timely manner.

The next section introduces the concept of a PE Roadmap. The Roadmap guides the process for the continued application of PE best practices at any stage in the development lifecycle. Continuity is facilitated by the reuse of PE Work Products developed along the way.

3 The Roadmap

The PE Roadmap provides the delivery mechanism for the PE Practice. The Roadmap outlines the steps required to build a toolset for continuous performance analysis, modeling, and capacity planning. The toolset provides a repeatable, reusable method to predict performance due to changes in applications, workloads and environments.

The Roadmap defines the basic steps to move from one lifecycle stage to the next and can be applied at any time during the software development lifecycle. The key to determining the steps lies in understanding the system as it exists today and creating a PE-path to the future, based on the goals of the business.

3.1 Building Blocks

To define the Roadmap, consider the types of information (i.e., building blocks) required for an application performance study:

- **Workload** – What types of transactions does the system process? What is the volume of transactions coming into the system?
- **Transaction Flow** – How are transactions processed by the system? What is their flow through the software components?
- **Resource Usage** – What are the computing resources consumed for each transaction? For example, CPU time, I/Os, memory and network message sizes.
- **Software Constraints** – How does the software constrain throughput of transactions? For example, how many threads, JDBC connections or parallel processes?
- **Environment** – What computing platforms, network components and topology comprises the system?

A ***model*** is the vehicle used to combine all of these elements to communicate and predict the performance, capacity, and dynamic behavior of the system. A model provides structure for the building blocks, a repository for performance data and a guideline for future analysis. When viewed from a

lifecycle perspective, a model has the following key characteristics:

- **Communication** – The model is a communication vehicle. It combines the key characteristics of an application and its supporting infrastructure for a true end-to-end view.
- **Prediction** – The application's behavior in terms of performance and capacity can be evaluated and predicted with the model.
- **Evolution** – The model is used to track the development of the application through the lifecycle. As the application moves closer to reality, so does the model; it is reused and refined at each stage of application development.

3.2 Quality of Information

At each stage of the lifecycle, we have access to some of the required information, but not all. During design, we have proposed flows from the architects and workload projections from the business owners. In development/test we can measure resource usage, software constraints and transaction flows. Once the system is in production, we know the environment and the actual workload.

The timeline shown in Figure 1 gives a more precise view of how the quality of information improves across lifecycle stages. Information begins with projections and improves toward actuals as the application moves toward production deployment.

The table in Figure 2 illustrates how the quality of information improves for the PE effort's building blocks across the development lifecycle.

From the table, we derive the definition of the Work Products and their applicability to different lifecycle stages; each cell equates to a Work Product for that stage. The performance model is the key Work Product, providing structure, continuity and predictive capabilities for the Roadmap.

3.3 Work Products

The term "work product" is used throughout the CMMI Product Suite [SEI2006] to mean

> "*... any artifact produced by a process. These artifacts can include files, documents, parts of the product, services, processes, specifications, and invoices. A key distinction between a work product and a product component is that a work product need not be engineered or part of the end product.*"

Here we use Work Products to represent artifacts, information and models developed at each stage of the lifecycle. These key pieces of information, performance models and data, contribute to the next lifecycle stage by providing a starting point and secondarily a checkpoint for the PE delivery based on the Roadmap.

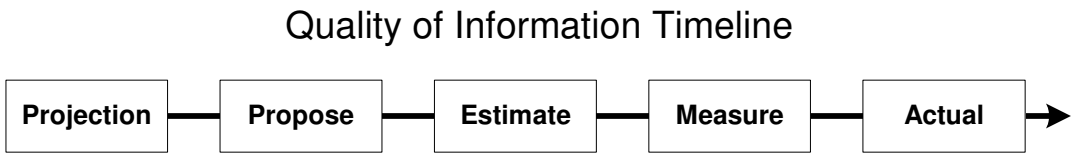

Figure 1. Quality of Information Timeline

	Design	Test	Production
Workload	Projected	Projected	Actual
Transaction Flow	Proposed	Actual	Actual
Resource Usage	Estimated	Measured	Actual
Software Constraints	Estimated	Measured	Actual
Execution Environment	Proposed	Measured	Actual

Figure 2. PE Building Blocks by Lifecycle Stage

Design Stage: In the design stage, we are able to characterize the following:

- Projected workload for the application, specified by the business.
- Proposed transaction flows, from the application architects.
- Proposed execution environment, hardware and network components.
- Estimated resource usage from our past experience with these types of applications or a "resources usage target" for each transaction.

We use this information to develop a baseline model of the design to predict the feasibility of the architecture – can it handle the expected workload based on our estimates? We typically use a high-level System Scalability model (see Stepwise Refinement in [SPEL2002]). The model and the information all become Work Products that can be reused and refined once the design is accepted.

Assume that we find the architecture acceptable and move on to develop the code. Once the application is ready for test, the PE Practice wants to reuse the Work Products from our design-stage study. How is this done?

Test Stage: In the test stage, we reuse the design stage Work Products as follows:

- Baseline performance model – the design model is updated with the actual transaction flows, resources usage measurements, software constraints, and test execution environment.
- Projected workloads – may be refined (by the business owners) and used again as inputs to our updated baseline model.
- Estimated resource usage – is compared to measurements to determine if performance budgets need to be refined.
- Results of design-stage performance analysis study – compared to test stage performance study, how accurate were our predictions? What can we do better next time?

Production Stage: Similarly, when we move to the production stage, we reuse the test stage Work Products in the following ways:

- Baseline performance model – the test model is updated with the actual resource usage, software constraints measurements, and production execution environment.
- Projected workloads – compared to actual workload to give feedback to the business – How accurate were their projections? How can we make more accurate projections in the future?
- Measured resource usage – is compared to actual resource usage from the production environment. We expect minimal changes.
- Results of test-stage performance analysis study – compared to actual production performance observed, how accurate were our predictions from test? What can we do better next time?

The Work Products support an efficient delivery mechanism by providing reusable tools and information; while the Roadmap guides the repeatable PE process.

Up to this point we've described the theoretical view of how the PE Practice delivers. Next, we demonstrate how we've applied these techniques in our business.

4 Case Study: Retail Banking Application

This study focuses on a retail banking application that is deployed to approximately 6,000 retail branch stores. This application supports the processing of banking transactions (e.g., deposits, withdrawals) in each branch. The mix of transaction volumes and variability of branch infrastructures adds to the complexity of managing the performance of these systems. The business owners asked the PE Practice to provide insight into performance and capacity of their retail application.

There were three phases of our analysis; each focused on one of the three key lifecycle stages:

1. **Evaluate Scalability (Test)**
   - What is the maximum sustainable transaction throughput?
   - What is the expected transaction response time as the load increases?
2. **Evaluate New Architecture (Design)**
   - Can a new implementation support the branch workloads?
   - How will the next generation application perform?
   - Should we upgrade to this new implementation?
3. **Ongoing Capacity Planning (Production)**
   - When do we need to upgrade our retail branch servers?
   - What size servers do we need?

We will walk through each of these phases to demonstrate how we applied Performance Engineering to meet the needs of the business by:

- Creating and reusing Work Products
- Following the Roadmap
- Delivering answers in a timely manner.

4.1 Phase 1 – Evaluate Scalability (Test)

Goals and Approach

In March 2005, the retail application development team constructed a load test environment, a subset of the current branch and backend data center infrastructure. The goal was to determine the capacity of the branches and predict transaction response times. Performance modeling was used to extrapolate from test to production.

The model was created from scratch as this was the first PE analysis of the application. Data collected in the test environment provided detailed transaction flows and measurements of resource usage and response times. The model was validated against response times, throughput and utilization from an actual production branch workload. Validation here describes the comparison of measurements from the production system to predicted response time, throughput and utilization results from the model. Results of the validation are show in Figures 3 and 4. Figure 3 shows the validation of the server CPU utilization and transaction throughput by comparing production measurements to model results. Figure 4 shows the response time validation results. Together these results demonstrated the accuracy of the model and gave us a high level of confidence in the model's predictive capability.

With a validated model in hand, the next step was to utilize the model to determine the capacity of the current configuration in terms of transaction volume and response time.

Scalability Results

The results of our modeling and analysis determined that a "typical" retail branch implementation of the application could handle up to 158 transactions per hour. Figure 5 shows the results of the scenario used to evaluate the capacity of a typical branch. Maximum throughput is reached at 21 tellers; at higher load levels throughput remains fairly constant and response time increases exponentially.

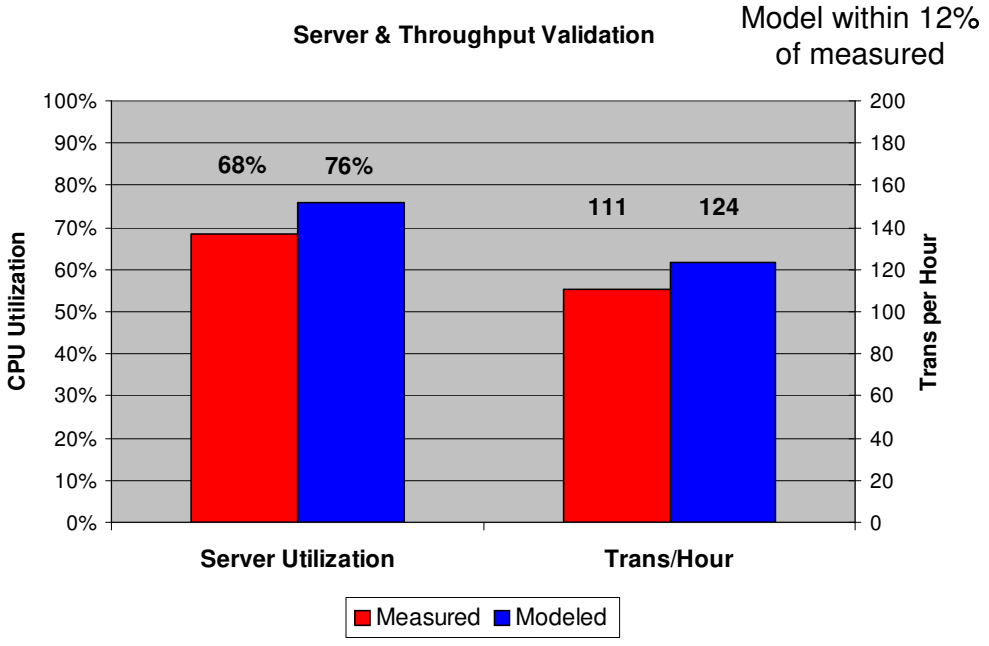

Figure 3. Validation Results - CPU Utilization & Throughput

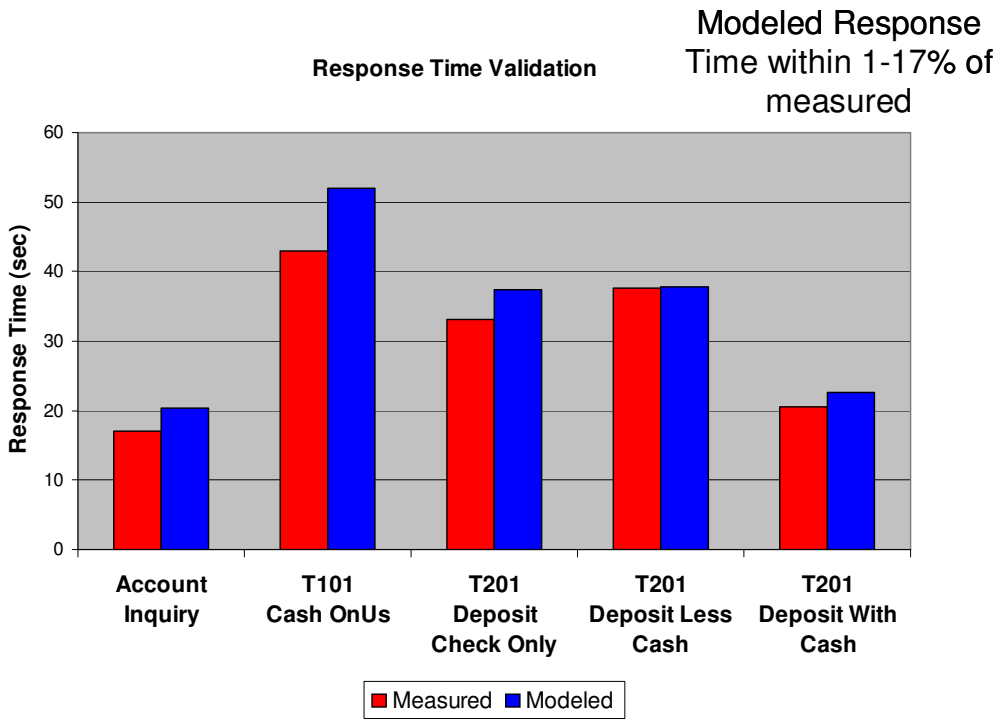

Figure 4. Validation Results - Response Time

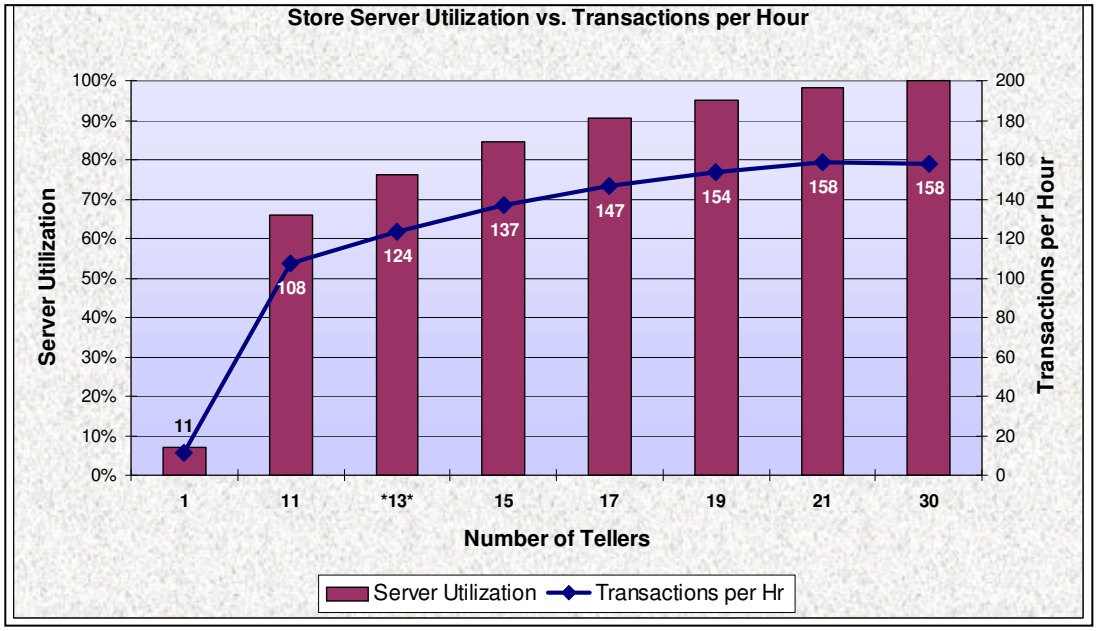

Figure 5. Scalability Results - Increasing Transaction Volumes

	Design	Test	Production
Workload	Projected	***Projected***	***Actual***
Transaction Flow	Proposed	***Actual***	Actual
Resource Usage	Estimated	***Measured***	Actual
Software Constraints	Estimated	***Measured***	Actual
Execution Environment	Proposed	***Measured***	***Actual***

Figure 6. Work Products – Phase 1 – Evaluate Scalability (Test)

Work Products Created

In this initial phase of our project, we created the following Work Products:

1. Validated baseline performance model which included:
   - Actual and projected workloads
   - Actual transaction flows
   - Measured resource usage and software constraints
   - Actual execution environment
2. Results of load test and modeling scenarios

These Work Products (highlighted in Figure 6) were leveraged in Phase 2 (next section) to improve the efficiency of our PE Practice and reduce delivery time.

4.2 Phase 2 – Evaluate New Architecture (Design)

The next step for the retail business required an evaluation of a new software architecture to upgrade and enhance the existing capabilities in the branches. The new software provides enhanced functionality utilizing Java, Smart Client, XML and sophisticated business rules. The cost of rolling out new code to all 6,000 branches would be high, so an analysis of the new application and architecture was critical to the bank.

Goals and Approach

The goals for the evaluation of the new architecture are to:

- Evaluate the new application for retail branches, both in terms of capacity and performance.
- Verify hardware estimates.
- Gain insight into the performance characteristics of the new application.

The approach followed that of the PE Roadmap and is a standard service of the PE Practice. It also utilized Work Products from the previous study. Specifically, we:

- Added the proposed application workflow to the model (overlaid on the original baseline model).
- Estimated resource usage of the new implementation. Interviews with the software vendor gave us our initial estimates. These estimates were then verified, refined, and validated using our knowledge of the current resource consumption for the application.
- Evaluated system capacity and performance using modeling scenarios with increasing transaction volumes.
- Verified proposed server upgrades and configuration changes.

Figure 7 displays the process followed for this phase.

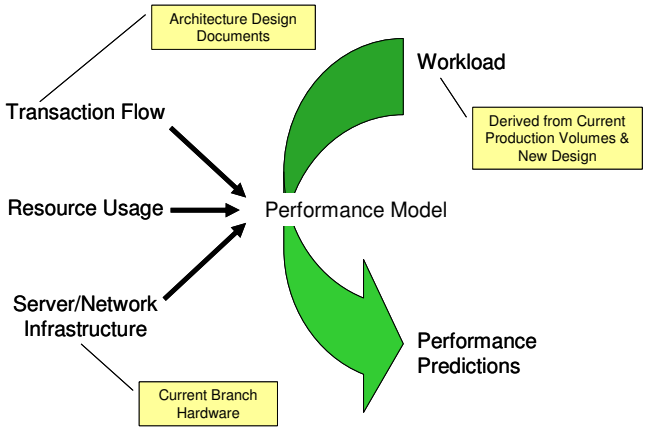

Figure 7. Process for Design Stage Modeling

Results

Analyzing results of the design for the new application required investigation and comparison of the following metrics:

- Response time
- Throughput
- CPU utilization

The scenario results (see Figure 8) showed that as the number of tellers increased, the CPU utilization on the branch server exceeds 90% at 20 users.

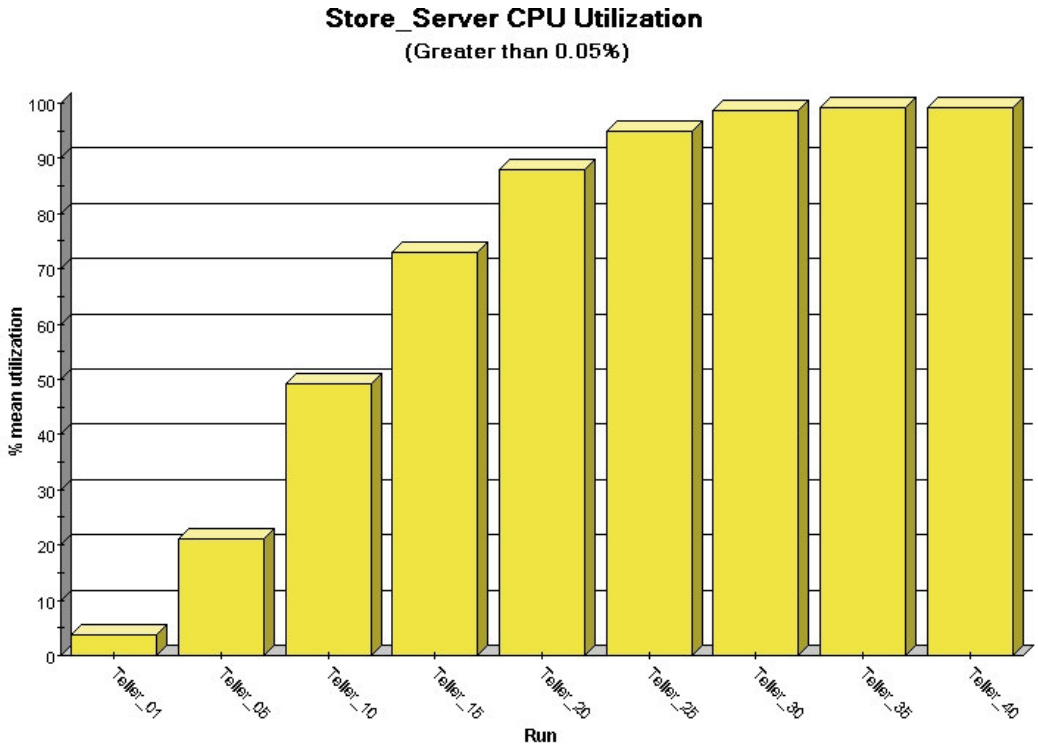

Figure 8. New Architecture - CPU Utilization Results

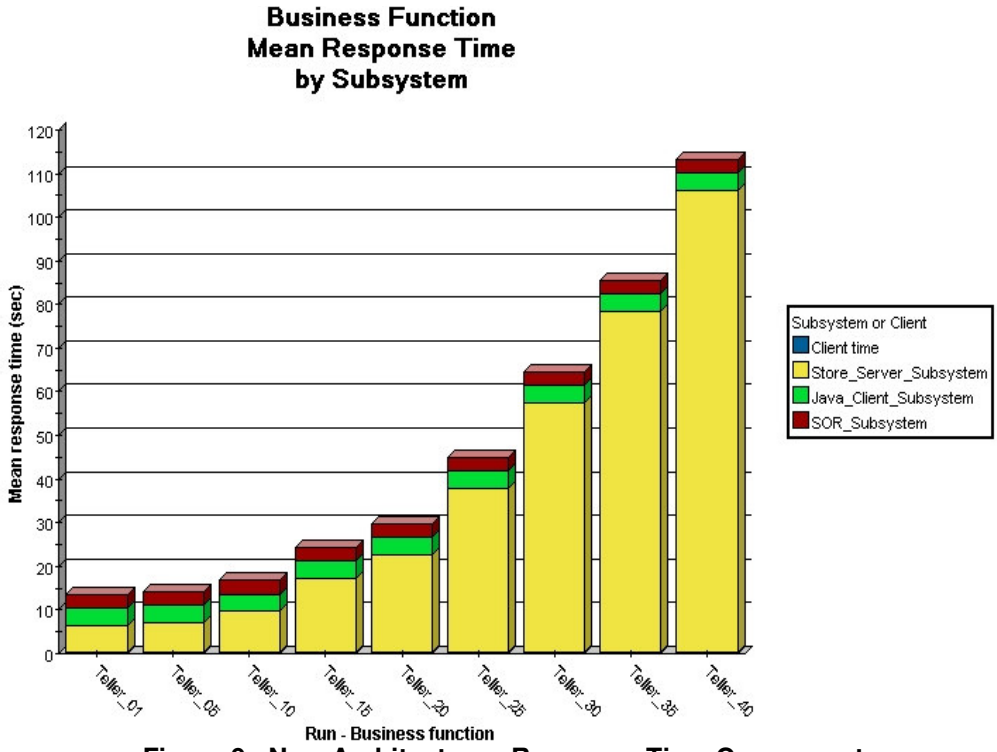

Figure 9. New Architecture - Response Time Components

	Design	Test	Production
Workload	***Projected***	Projected	Actual
Transaction Flow	***Proposed***	Actual	Actual
Resource Usage	***Estimated***	Measured	Actual
Software Constraints	***Estimated***	Measured	Actual
Execution Environment	***Proposed***	Measured	Actual

Figure 10. Work Products – Phase 2 – Evaluate New Architecture (Design)

This utilization bottleneck causes the response times to exceed today's response times; and can be seen in the response time breakdown. Figure 9 shows the response time components across the tiers in the new application architecture. The knee in the response time curve kicks in at about 20 tellers; adding additional tellers will result in severely degraded response time.

Work Products

In this analysis of a new design for the application, we utilized the following Work Products from our Phase 1 analysis:

- Baseline performance model of the current infrastructure.
- Resource requirements to provide a reasonableness check against the new application's estimates.
- Actual and projected workload volumes.

In addition to the Work Products from Phase 1, Figure 10 shows the new design stage Work Products created from this phase. It should be clear that we are gradually filling in the Work Products grid for this application across the entire lifecycle.

In the process of evaluating the new application technology, both from a business perspective as well as a performance perspective, the business chose to postpone the new implementation. To support the business, the PE Practice must continue its focus on predicting the capacity of the current production system. When will hardware upgrades be required?

4.3 Phase 3 - Ongoing Capacity Planning (Production)

The business has chosen to stay with the current retail application for the foreseeable future. As such, we now turn our attention back to the current infrastructure and how we can utilize our accumulated Work Products to implement ongoing capacity planning. The business expects to purchase additional hardware to support increasing transaction volume, as this is deemed to be a cost effective approach for the near term. They need to know when the hardware upgrades will be required.

Goals & Approach

The PE Practice was asked to provide regular capacity planning reports and to predict when hardware upgrades will be required. For this stage of our work, the goals were to:

- Provide monthly capacity reports for the branches
- Predict when the hardware will reach capacity
- Determine the effect of hardware upgrades

A key success factor for ongoing capacity planning is efficiency; the initial analysis and model development took about 3 weeks, much too long for practical continuous capacity planning. Our capacity planning solution must provide analysis in hours, not weeks. Plus, the large number of branches (6,000) drove the need for a more streamlined methodology.

The need for efficiency dictated the use of an analytic modeling approach and an automated data collection process to update the models with current resource usage and workloads.

Our approach included:

- Implementing an automated collection of resource usage and workload data from the branch and data center servers.
- Creating analytic models of the retail branch application based on current activity.
- Reporting results to the business including resource utilization and response times in a consistent format.

For our initial analysis, we were asked to evaluate the effect of increased workload growth by 5% per month for 14 months.

Results

The PE Practice provided the results in approximately one day and implemented the capability to produce reports monthly.

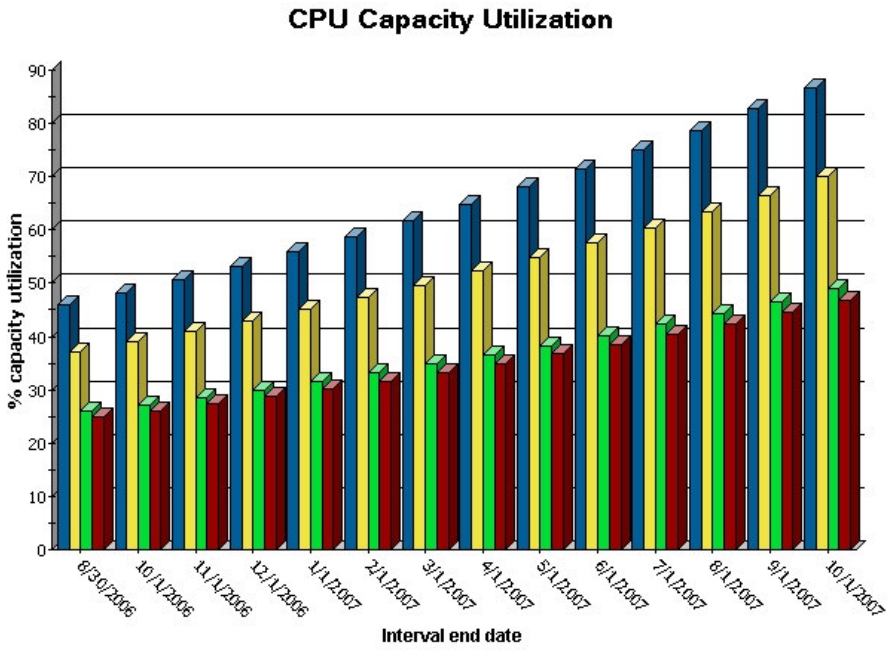

Figure 11 – CPU Utilization Projection – Current Hardware

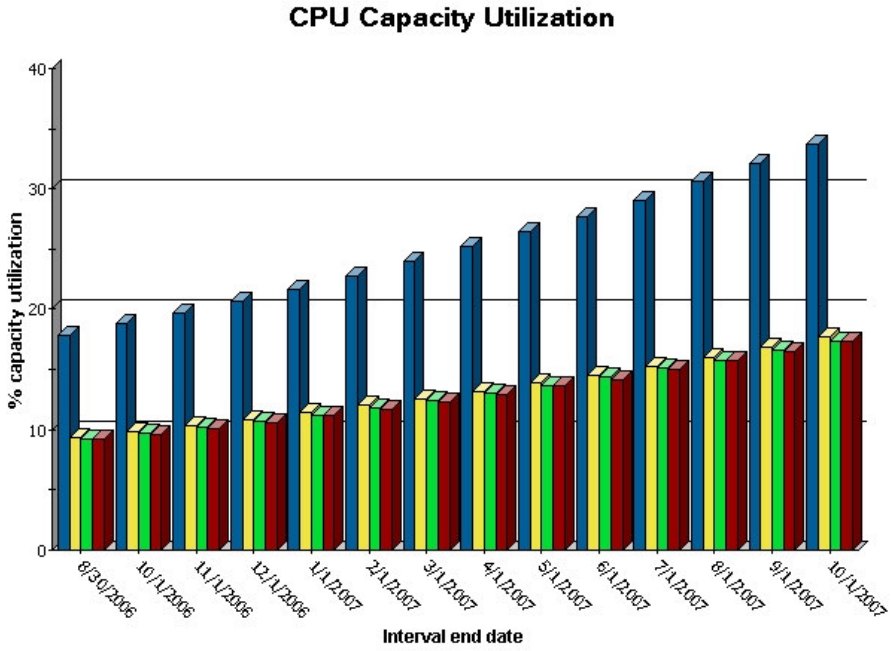

Figure 12 – CPU Utilization Projection – Upgraded Hardware

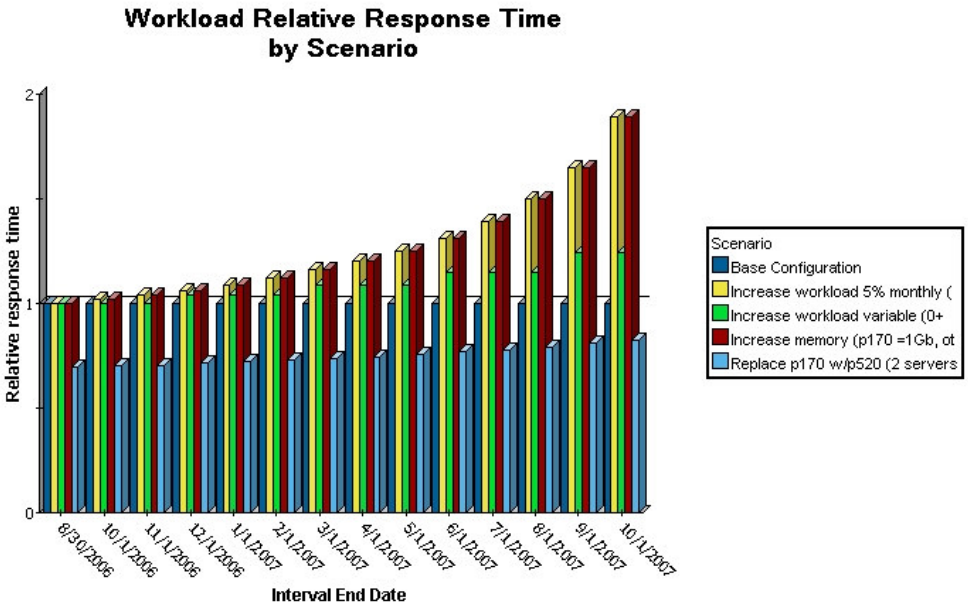

Figure 13 – Device Utilization Projection

	Design	Test	Production
Workload	Projected	Projected	Actual
Transaction Flow	Proposed	Actual	***Actual***
Resource Usage	Estimated	Measured	***Actual***
Software Constraints	Estimated	Measured	***Actual***
Execution Environment	Proposed	Measured	Actual

Figure 14. Work Products – Phase 3 – Ongoing Capacity Planning (Production)

The study results showed that the expected monthly workload growth of 5% would exceed the capacity of the current hardware (IBM p170's) in approximately 12 months.

Figure 11 shows the CPU utilization results for the increase. Note that the servers are not load balanced and one in particular hits the wall first at over 80% by September 2007.

A second scenario evaluated upgrading the servers to p610 models. The results showed that the upgraded servers could sustain acceptable utilization for the increased load as shown in Figure 12.

Additional scenarios evaluated included:

- What is the effect of changing the workload by variable percentages (not fixed at 5%, but 3%-10% increases monthly)
- How does increased memory improve the capacity of the servers
- Replace p170's with p520's

Figure 13 shows the results of these scenarios.

Work Products

In this phase, we reused past Work Products and created new ones. The Work Products that we utilized from previous phases:

- Workload projections
- Execution environment
- Resource usage data sources

The new Work Products include:

- Automated resource data collection
- Analytic models of retail branch servers
- Capacity reports

Figure 14 is updated to show the full compliment of Work Products from all three phases. We now have a complete set of Work Products that can be used for full lifecycle Performance Engineering of the retail banking application.

5 Conclusion

We've seen through specific examples how a PE Practice can deliver services with reusable Work Products at different stages of the lifecycle; even in the case where the process starts at production and moves backwards to evaluating new design alternatives.

Performance Engineering provides a Roadmap for the PE Practice to deliver and build a continuous, repeatable process for applying best practices. Additionally, the Work Products provide a basis to

- Efficiently deliver PE services at any stage in the lifecycle
- Reuse tools, techniques and information
- Apply PE techniques and methods more effectively

Performance Engineering can and should be used across all stages of the application development lifecycle. As practitioners we need to support PE in all lifecycle stages and understand how Work Products are not merely an end result; instead, they are a stepping stone to the next stage.

6 References

[GIMA2004] Richard L. Gimarc, Amy Spellmann, and Jim Reynolds, "Moving Beyond Test and Guess - Using Modeling with Load Testing to Improve Web Application Readiness", CMG 2004 International Conference.

[LETN2005] Charles Letner and Richard Gimarc, "A Methodology for Predicting the Scalability of Distributed Production Systems", CMG 2005 International Conference.

[SEI2006] Carnegie Mellon Software Engineering Institute.

[SMIT1990] Connie U. Smith, "Performance Engineering of Software Systems", Addison Wesley, 1990.

[SPEL2000] Amy Spellmann and Richard Gimarc, "eBusiness Performance: Risk Mitigation in Zero Time (Do It Right the First Time)", CMG 2000 International Conference.

[SPEL2002] Amy Spellmann and Richard Gimarc, "Stepwise Refinement: A Pragmatic Approach for Modeling Web Applications", CMG 2002 International Conference.

[ZAHA1993] Bill Zahavi, "Modeling the Performance Budget (A Tutorial)", CMG 1993 International Conference.

[ZAHA1995] Bill Zahavi, "Modeling the Performance Budget - A Case Study", CMG 1995 International Conference.