technical concept

# Taking the risk out of rapid application deployment

Integrating Application Profiling into the Application Deployment Process to Ensure Applications that Perform

Creating and maintaining competitive advantage in today's wired business world requires shortened product development cycles, frequent updates and more complex integration. Couple these factors with the increased business exposure inherent in "customer-facing" applications and it becomes obvious that the level of risk from poor application design and deployment has risen dramatically.

Developers used to build and deploy distributed applications using a "waterfall" methodology: definition, design, building, testing and deployment following sequentially. This necessarily limited the analysis of performance, leading to a lengthy, trial-and-error testing phase or requiring risky shortcuts to meet delivery schedules.

Once testing in the development lab ends, the application is put into production. This step is often described as "throwing it over the wall" for the IT or network operations staff to deploy, terminology which expresses the typically poor communication between the development and network teams.

The shortcomings of this process are obvious and severe: just because the application works in the single-user QA lab doesn't mean it will work in a multi-user, distributed production network. Waiting to discover performance problems until deployment (you might call it "performance testing by the end user") leaves painfully few options: live with the poorly performing application, stop deployment and go back to rework the code, or scrap the project. Each alternative represents missed business opportunities and significant expense. The conclusion is obvious: the earlier you catch and fix performance problems, the easier it is and the less the cost will be.

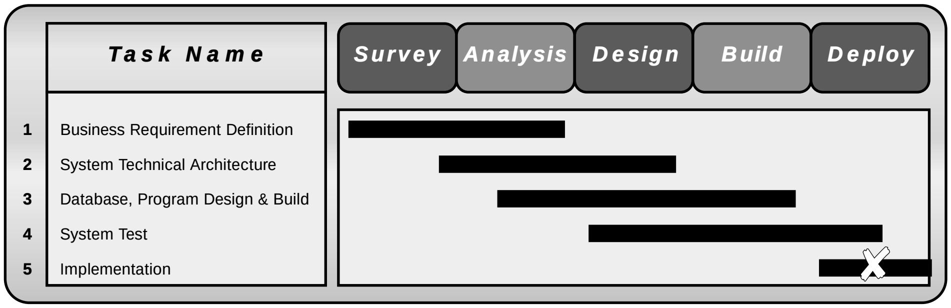


Figure 1. Failure to test for performance
*(X is where performance problem is discovered)*

### Why didn't the application perform properly?

Some deceptively obvious technology scapegoats now rear their heads. Perhaps the application or database server can't support the demands of many concurrent users. Or maybe the network doesn't have the bandwidth required to effectively carry all of the new application traffic. After all, the difference between the QA Lab (where the application performed well) and the production environment (where it is failing) is simply the number of users and the network itself. Since both of these are perceived to relate to capacity (of the server or of the network), planning for adequate capacity seems an appropriate solution.

Load testing solutions abound, hardware vendors continually update performance benchmarks, and network modeling tools can be used to predict the performance of various network configurations. But capacity planning alone doesn't ensure successful application performance. While the tools may deliver accurate and valuable information about the system components they test, they often fall short on two accounts: the metrics used and the integration of the system components.

Many capacity planning tools use "internal" (network or server) metrics to gauge projected performance, but the most meaningful measure of production readiness is the end-user experience; for performance testing, this is the end-user transaction response time. And testing client/server components separately and in isolation is no guarantee that they will perform well in an integrated system, which includes not only servers and clients, but the service qualities of the production network as well. Capacity planning alone does not effectively address this integrated system.

## Server load testing

To help address the problem of application deployment failure, project managers have incorporated additional planning and analysis into the development life cycle. They quantify server performance requirements for multi-user systems using load testing methodologies and tools, typically as part of a comprehensive QA testing strategy. In a waterfall development methodology, this is performed at the end of the life cycle. Load testing simulates, or synthesizes, the real-world multi-user demands on distributed systems, including hardware, software and application logic. This is accomplished by capturing transactions and replaying them according to user-defined workloads, including user populations and transaction mixes, or by leveraging existing PCs to replay the transactions as real users. Load testing methodologies incorporating these tools promise successful application deployment—from the server's perspective.

## WAN capacity planning

Client/server networks exhibit chaotic behavior, make wildly varying traffic demands, and use multiple application protocols, adding to the difficulty of capacity planning. From a network perspective, remote connections over wide-area networks are not only the obvious limiting performance factor but also represent the most significant network infrastructure costs. WAN capacity planning therefore has become a required step in any application deployment methodology.

Simple spreadsheet calculations of bandwidth requirements have been used successfully for many years to design private-line terminal networks as well as voice networks. However, distributed computing is more complex, and the two-dimensional spreadsheet no longer suffices to analyze traffic on a multi-dimensional system.

A few modeling and simulation vendors adapted their engineering simulation languages to a network paradigm. These tools use techniques such as discrete-event simulation and analytical modeling to analyze the performance of devices, media and protocols. Discrete event simulation represents traffic as a sequence of messages; the movement of this traffic through the network model is simulated by keeping track of the state of the network devices as the traffic is "stepped" through the model, one event at a time. Analytical modeling uses a number of equations (such as queuing theory and load calculations) to describe the network's performance.

These tools were frequently too complex and time-consuming to use, and suffered from drawbacks inherent in their design: the emphasis on modeling network device performance characteristics and network protocol behavior distracted from the relatively straightforward goal of WAN capacity planning. They ignored the most significant source of error—describing user behavior.

Application deployments continued to fail, even with effective server load testing and network capacity planning. Why? The fault must lie in either the approach to testing or the test tools themselves. Improving the accuracy of the test tools misses the point; it simply adds to the complexity of the test. Addressing the approach to testing can deliver tangible benefits, albeit at a high cost, as we review next.

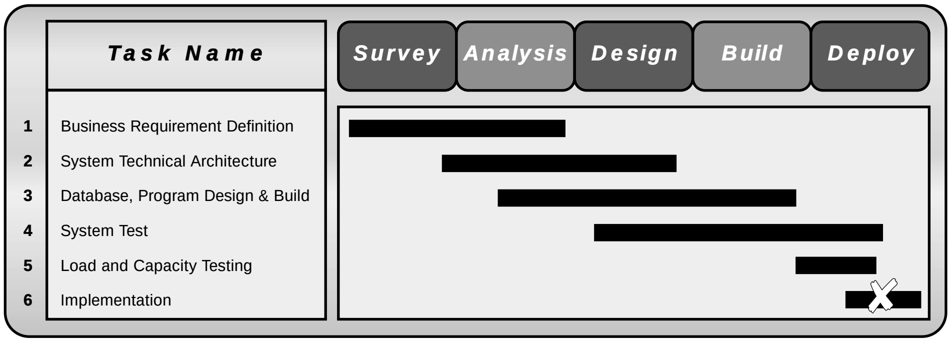


Figure 2. Inadequate Pre-deployment Performance Testing

### Application pilots

Pilots, often used for end-user acceptance testing, can partially fill in the gap in deployment testing methodologies. Once load testing has been completed and the appropriate capacity planning done, piloting application usage in a production network can expose unforeseen performance issues. For the first time the integrated system (including the network) can be tested as a whole, with the end-user response time as the key performance metric. Piloting often involves multiple phases; perhaps first among local users at the corporate headquarters where IT staff can easily monitor the results, later at one or more remote sites.

The combination of load testing, capacity planning and application pilots can, in most cases, identify applications that will not perform according to end-user expectations. We are left, however, with three significant and costly concerns:

- as development schedules slip, testing time may be cut back
- uncovering a performance problem after QA testing leaves little time to correct the problem
- discovering a performance problem in this manner provides little input for analysis to aid in fixing it.

Under extreme pressure to correct the problem, the test results themselves may become suspect. These pressures often result in “knee-jerk” reactions—adding WAN bandwidth is a common example. Unfortunately for many business initiatives (and network careers), bandwidth often is not the answer.

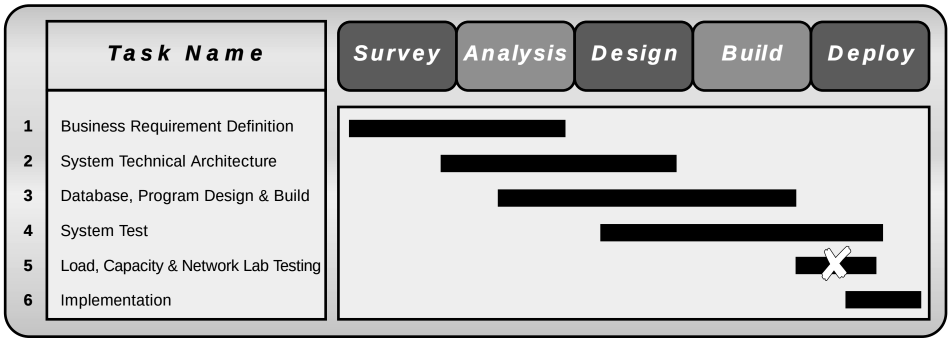


Figure 3. Network-centric Application Review

### Network integration labs

The interaction between the production network and the application is the piece of the puzzle left untested prior to the production pilot. In response to this problem, many companies have built and staffed sophisticated network integration labs. These labs duplicate key hardware and media components of the production network, including switches, routers, simulated or real WAN circuits, and frame-relay connections, often including clients and servers. Sometimes included in these labs are hardware “black boxes” which emulate the bandwidth and latency characteristics of WAN circuits. Network integration testing within the deployment process helps uncover response time failures, but makes the application development life cycle longer.

Network integration testing, however, still fails to provide feedback to the development team. The results are most often of a binary, pass/fail nature, quantifiable but unqualified and uninterpreted. If a particular transaction will take too long to complete, where is the problem? These are complex integrated systems. Is the problem the network? The database? The application server? The application architecture? Will additional capacity solve the problem and meet end-user performance goals? In short, what physical characteristic or logical behavior should be corrected?

Many Fortune 500 companies have built test facilities, often at great expense, staffing them with project leaders and network planners, in an attempt to “bridge the gap” between the systems and applications teams and the network and IT staff. Most of these labs fail to deliver on their promise to objectively profile application performance from a business perspective while adding real value to the process. The result? Failure to cut the risk of deploying an application that performs poorly.

### Value-added application deployment planning

The good news is that these integration labs can quickly and easily be turned into value centers by addressing the root cause of their failure—the lack of a deployment test solution that can deliver appropriate and valuable information in a timely manner. This information can be thought of as business performance decision analysis—the ability to relate technology decisions directly to the business benefits. Business performance decision analysis answers many fundamental questions, including:

- Will this application be able to meet the end-user (business) performance requirements?
- How can I ensure that my infrastructure can support these requirements at deployment time?
- Which component of the integrated system will be the bottleneck once the application is deployed?
- How can I architect or tune that component, and what will be the resulting improvement in end-user response time?
- How much will the desired performance cost?

This step in the deployment planning process is often referred to as application profiling or transaction performance baselining. Application profiling gauges the performance of the integrated system in both network and application terms. For the first time, developers can visualize and quantify the network-related behavior of the application, while network analysts have a graphical view of the application components' execution across the network. Once the integration lab is able to provide this information consistently, it begins delivering true business value. Look at the primary benefits of a successful application deployment planning function:

- Business unit managers will benefit from the services because the information delivered removes much of the remaining risk and exposure of deployment failure.
- Developers will benefit because the service will enable them to quickly understand the components of distributed application performance, the performance implications of architectural decisions, the impact a production network will have on their application, and the resulting opportunities for improving their code.
- Network planners will benefit from the service because the performance aspects of new applications will be understood and quantified prior to deployment, and infrastructure costs to deliver the required quality of service can be allocated "just in time," evaluated according to the business benefits.
- Network managers will benefit from the stored knowledge represented by the baseline information. This data forms the foundation for "rapid application performance troubleshooting," where a profile of baseline or normal application behavior can be easily compared with a profile of the degraded performance, quickly identifying the component responsible for the reduction in service.

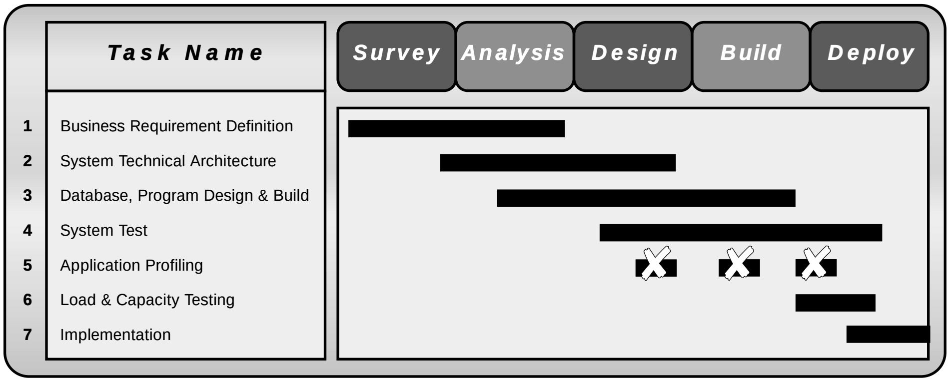


Figure 4. Application Profiling

## Evolving application performance engineering

Today, many development organizations are adopting iterative development methodologies emphasized by Rapid Application Development (RAD) cycles. Unlike waterfall development life cycles, where testing is done at the end of the project, iterative life cycles specify testing at multiple points during development. It is easy to understand the importance of including system performance analysis and predictive tuning in the process. Identifying and addressing flaws early, especially performance limitations, cuts the cost to fix them and minimizes the impact on the project schedule.

System performance testing can take two parallel approaches—measured and predictive. System performance analysis and tuning is most frequently accomplished by measuring with various QA and load-testing tools. Performance problems induced in the test lab can be readily identified and addressed. Predictive performance tuning, on the other hand, means tuning the application for the intended deployment environment (rather than for the test environment).

Attempting iterative testing of an application for performance in a distributed environment using traditional methods would simply exacerbate the problems, identified earlier, associated with monolithic testing at the end of the development life cycle. It would require either multiple, time-consuming “mini-pilots” using the production network or multiple trips to a complex, elaborate test lab. Again, the results of these tests would be of little analytical value to the developer. But once the integration lab implements a proven application profiling process, this type of performance testing becomes an important part of the development process.

As developers and their managers become more sensitive to designing and building performance into their applications, and more aware of the value of predictive tuning, it will become important to bring some components of network performance testing into the development lab. The development team will begin to profile the application’s performance themselves, during the appropriate iterative testing phases.

## Bibliography

Clewett, Annette. *Network Resource Planning for SAP R/3, BAAN IV, and PeopleSoft.* McGraw-Hill, 1998

Edwards, Jeri. *3-Tier Client/Server at Work.* John Wiley & Sons, 1999

Fournier, Roger. *A Methodology for Client/Server and Web Application Development.* Yourdon Press, 1998

Danielson, Dave. *"Just-In-Time Network Design."* An Acuitive, Inc. White Paper, 1999

The Hurwitz Group. *"Business Benefits of Performance Testing."* A Compuware White Paper, 1999

McConnell, John. *"Performance-centered Application Design and Deployment."* An Optimal Networks White Paper, 1999

## Delivering applications that perform

Delivering applications that perform requires a life cycle approach to understanding the performance of the integrated system. Compuware provides rapid application deployment solutions that integrate with any current development and deployment process, and is the only vendor with a true application profiling product in addition to traditional load testing and capacity planning solutions. These solutions provide rapid cross-functional insight into the complexities and interactions of multi-tier, multi-threaded distributed applications. Successful application deployment requires that development and network teams work together; Compuware's solutions provide the analysis and communications vehicle that makes that happen.

## Compuware products and professional services—delivering quality applications

Compuware is a leading global provider of software products and professional services that help IT professionals develop, integrate, test and manage the performance of the applications driving their businesses. Our solutions enhance every step in the application life cycle—from defining requirements to supporting production levels—for web, distributed and mainframe platforms. Our services professionals work at customer sites around the world, sharing their real-world perspective and experience to deliver an integrated, reliable solution.

Please contact us to learn more about how our comprehensive solutions can help your organization improve productivity, create higher quality applications and speed time to market.

COMPUWARE®

www.compuware.com





601 • 7/01