Realtime
publishers

The Shortcut Guide™ *To*

Assuring Website Performance through External Web Monitoring

sponsored by

neustar™

Dan Sullivan

Copyright Statement

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology eBooks and guides from Realtime Publishers can be found at http://nexus.realtimepublishers.com.]

Chapter 4: Implementing External Monitoring: A Roadmap from Building the Internal Business Case to Long-Term Maintenance

Establishing and maintaining an external monitoring system for business applications is becoming increasingly important to supporting strategic initiatives. The best made plans for deploying a new product or service line will fall flat if customers cannot readily complete transactions or depend upon a reliable service. Even loyal customers will switch providers if technical impediments, such as slow and unresponsive Web sites, get in their way. Customers are in many ways already performing external monitoring on your business applications. The problem with this is that feedback from customers too often manifests itself in declining revenues and increasing churn.

This guide has examined several factors that are driving the adoption of external Web monitoring, including increasing expectations for service delivery, increasing application complexity, and demands for greater visibility into networks and applications. Effective external monitoring combines geographically distributed agents, application-specific monitoring tests, and reporting services that enable rapid detection of and response to performance problems. We have also discussed how external monitoring can help acquire customers, control operational costs, and protect brand image. This concluding chapter describes a framework for implementing external monitoring. This framework addresses several key areas:

- Establishing the business case for external monitoring
- Planning an external monitoring operation
- Designing and implementing external monitoring
- Developing testing and acceptance criteria
- Communicating and responding to events
- Developing governance and policy
- Performing long-term maintenance

As these topics show, it is essential to address both business and technical considerations with external monitoring. Let's begin by considering the business case for external monitoring.

Establishing the Business Case for External Monitoring

We monitor to maintain business services. If services are down or underperforming, there can be significant and demonstrable impact on revenues. External monitoring provides detailed information on performance levels that allows systems and application administrators to efficiently isolate and correct problems. That is the short version of the business case for external monitoring.

> **Performance Impact Studies**
>
> For more information about the correlation between performance and revenue, see "Maintaining and Growing Revenues" in Chapter 3, which describes studies by Google, Microsoft, and Shopzilla.com on this topic.

For a more in-depth look at the business case for external monitoring, we will consider four steps to justifying an investment in external monitoring:

1. Evaluating the status quo
2. Identifying weaknesses in current service delivery platforms
3. Assessing the impact on business performance metrics and service level agreements (SLAs)
4. Estimating return on investment (ROI)

These steps are designed to identify business risks in current operations and show how external monitoring can mitigate those risks.

Evaluating the Status Quo

We may be tempted to jump right in with external monitoring and start thinking about where to deploy it and what type of tests to run to measure performance. The temptation is understandable, but it would be a mistake to proceed without a clear understanding of baseline performance of key business systems.

Inventory Services

We should start with an inventory of business systems that make use of the Web. Some of the most obvious are:

- Corporate Web sites
- Online product catalogs and sales services
- Employee portals
- Business partner services
- Customer support Web sites

Also include application programming interfaces (APIs), many of which are provided through Web services. Business partners, for example, might use your business Web services to enhance their Web site functionality or to enhance their process flows by directly incorporating your services. As a general rule, if an application is accessible by a Web address (for example, URL, URI) and the application is part of a business process, it should be included in the inventory. A Web service that is accessed only by internal services should still be included; it might be used by externally accessible systems today or might itself someday become externally accessible.

Collect Performance Data

For the most critical business systems, we should next collect basic performance data. In many cases, we will have collected only basic operational performance data such as:

- CPU and memory utilization rates at the operating system (OS) level
- Data sizes and growth rates from database management systems
- Application downtimes from log files
- Network utilization and latency from network monitoring tools

Collecting this data will help us understand how much of our computing, storage, and network capacity is used and how reliably our applications are running. Presumably, we do not have an external monitoring system in place, so our performance data is limited to what we collect internally. Nonetheless, this is valuable data for building our business case. For example, with this data, we can answer questions such as:

- How much capacity is available in existing servers?
- Are some existing SLAs being met (without external monitoring, we cannot tell whether all SLAs are met)?
- Are there obvious patterns associated with application downtime?
- If there are known performance problems, is there an obvious source of the problem in a single application or service or is it more likely a combination of factors that contributes to a performance issue?

At this point, we do not need to do an in-depth analysis of particular performance problems. The goal here is to create a high-level overview of the state of performance on key business systems. This perspective provides the foundation for the next step in establishing the business case: identifying weaknesses.

Identifying Weaknesses in the Current Service Delivery Platform

Customers see the impact of weaknesses in service delivery platforms in slow performance and unreliable systems. When servers run at near-peak CPU and memory utilization, there is little room to accommodate increases in demand. A similar risk occurs with networks operating with little excess capacity. Customers might see this as “on again/off again” performance, which could lead to uncertainty about using a service. This is especially a problem if a business partner is depending on one of your services as part of a multi-step workflow. Consistent performance saves customers and business partners from having to plan on a workaround if your service is not performing as needed.

The most severe form of performance problems is when a service is completely unavailable. This can happen for a number of reasons, ranging from a hardware failure to a critical bug in an application. Load balancing across a cluster of servers or other failover techniques can mitigate the risk of hardware failures. Sometimes the worst effects of software errors can be prevented by avoiding the triggering effect of the error, such as low memory in an application that does not properly manage memory utilization.

These mitigating strategies can have unexpected consequences for performance. For example, the failure in a server in one data center may cause traffic for an application to shift to another data center. The additional demand on the failover server then leads to increased transaction times for customers, which translates into an increase in incomplete transactions, abandoned shopping carts, and other measures of lost business.

Impact on Business Performance Metrics and SLAs

Once again, without external monitoring in place, we cannot get a complete picture of application performance and its impact on business performance. However, we can at least frame an outline of how performance affects SLAs.

In the previous steps, we collected technical data to assess performance; now, we collect data from a business perspective. Business performance metrics can vary widely but could include:

- Number of completed transactions in a given period of time
- Gross revenue from sales by time period, region, product category, and so on
- Number of problem tickets or customer complaints
- Number of new leads generated
- Number of new customers acquired/existing customers lost

These metrics can be especially helpful in identifying trends related to performance. For example, if the number of customer complaints or lost customers correlates with performance and reliability problems, that would constitute strong evidence for the need for improved monitoring and response.

SLAs are typically based on important performance metrics, such as application response time and application uptime. Thus, how well SLAs are met can be a good indicator of weaknesses in the current service delivery platform.

With an inventory of existing services and performance data associated with those services and a high-level overview of weaknesses in the current platform, including SLA criteria, we can move on to estimate the business benefits of improving monitoring and response.

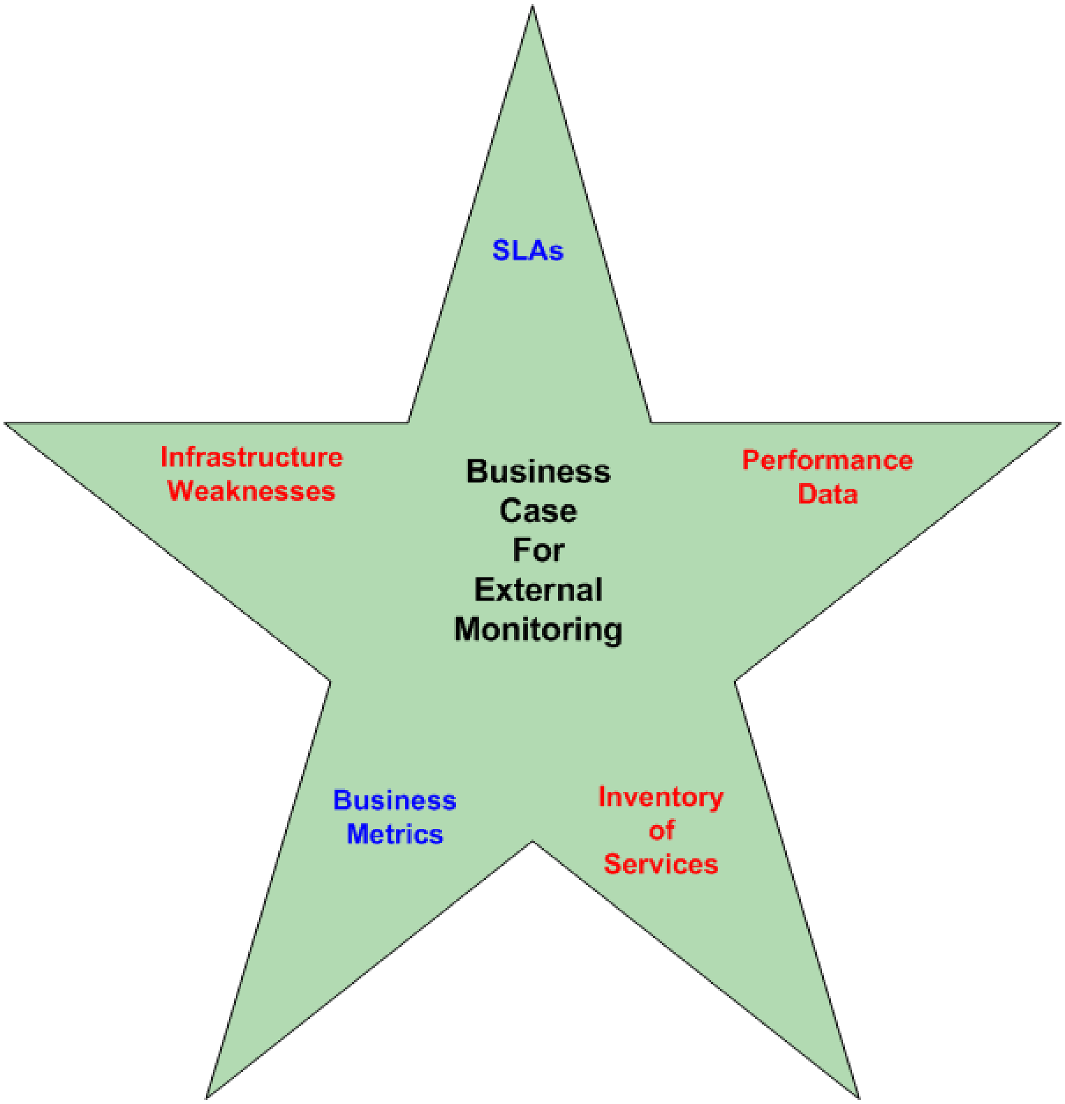
SLAs
Infrastructure Weaknesses
Business Case For External Monitoring
Performance Data
Business Metrics
Inventory of Services

Figure 4.1: The business case for external monitoring is built on a combination of business metrics (in blue) and technical metrics (in red).

Estimating ROI

In spite of whatever weaknesses might exist in current business systems, how are we to estimate the business benefit of proposed changes, such as implementing external monitoring? This is probably the single most challenging question in IT. Few topics will generate as much debate as the best way to calculate ROI or comparable measures for making technology investment decisions. It is safe to say entire books could be written on this topic; the best we can hope for here is to outline one approach that addresses external monitoring as specifically as possible.

ROI is a function of costs and benefits, in the form of savings and increased revenues. External monitoring providers can offer cost estimates for monitoring services. To that we would want to add the cost of internal support, both during the rollout and during operational use. Some of the potential internal costs include:

- Staff time to write test scripts
- Time to deploy scripts to remote agents, although most providers address this for you
- Management time to define policies and procedures
- Management review of summary reports and trend analysis

We are not including the time associated with responding to performance issues because those costs exist regardless of whether external monitoring is deployed. This exclusion, however, runs the risk of underestimating the benefit of external monitoring. Problems that are detected earlier are often solved at lower costs than if they were detected later. For example, detecting poor performance on one set of servers and rerouting traffic can save some number of abandoned shopping carts or partially completed transactions. How would we calculate the lost revenue that could have been saved with external monitoring?

Let's start by saying there is no single, best calculation for all situations. Given that, here is one reasonable approach to calculating the increased revenue due to reduced cart abandoning:

1. Using log data from Web site logs, select a period of time in which services were available with high-performance rates.
2. Reviewing transaction data during this period, determine the number of times products were added to a shopping cart but the purchase was not completed. This is the baseline abandonment rate.
3. Using internal monitoring data, select several other periods of time in which performance was degraded.
4. For each of those periods, calculate one or more measures of performance relative to the high-performance period identified in step 1. Examples of such measures are the decrease in the number of transactions and the increase in the number of abandoned carts.
5. For each measure, calculate the average lost profit per unit, such as the lost profit due to transactions that were never started or lost profit due to transactions that were started but not completed (for example, abandoned carts).

Assuming we have a reasonably representative sample of degraded performance periods, these calculations should give us some idea of financial losses due to poor performance.

A simplified example is provided in Tables 1 and 2. To keep the math obvious, assume that each transaction has a marginal profit of $10. In this example, a set of five periods with poor performance cost at total of $19, 850.

	Number of Completed Transactions	% Decrease in Transactions	Lost Profit Due to Loss of Transactions ($10/Transaction)
High Performance Period	2000		
Low Performance Period 1	1800	10%	$2,000
Low Performance Period 2	1900	5%	$1,000
Low Performance Period 3	1750	13%	$2,500
Low Performance Period 4	1400	30%	$6,000
Low Performance Period 5	1500	25%	$5,000
		Total Losses By Type	$16,500

Table 4.1: If we assume 2000 transactions/unit of time during periods of high performance and an average marginal profit of $10, we can calculate the cost of degraded performance in terms of lost transactions.

	Number of Abandoned Carts	% Increase in Abandoned Carts	Lost Revenue Due to Abandoned Carts
High Performance Period	100		
Low Performance Period 1	140	40%	$400
Low Performance Period 2	130	30%	$300
Low Performance Period 3	170	70%	$700
Low Performance Period 4	185	85%	$850
Low Performance Period 5	210	110%	$1,100
			$3,350

Table 4.2: Similarly to estimating lost transactions, we can calculate the cost of transactions that are started but never completed.

As a final step in calculating the benefit of deploying external monitoring, you might want to apply a discount factor when calculating the potential for recovering revenue. External monitoring will not eliminate all performance problems (hardware will still fail), but it can help us resolve problems faster and detect potential problems more quickly. Judgment is called for here. Will faster resolution avoid 40% of the potential loss or 75% or more? What is the probability that our guess is correct? We cannot answer these questions with confidence, so it may be best to consider three possibilities: best case, expected case, and worst case scenarios and adjust parameters accordingly.

This approach to calculating ROI attempts to recognize the difficulties in making necessary estimates. It does assume that past performance indicators are good measures of future performance. The validity of this assumption will depend on how many other factors are changing in the market as well as in the application delivery platform. The calculation may underestimate the potential upside of an investment in external monitoring. For example, with improved monitoring, a business may be able to increase key business metrics well above the industry standard, providing a competitive advantage that did not exist prior.

One can readily make the business case for external monitoring in spite of the difficulties with ROI calculations. By evaluating the status quo, identifying weakness in the current service delivery platform, and assessing the impact of performance in terms of business metrics and SLAs, we can formulate a foundation of the best available data for making business decisions. By acknowledging the limitations of ROI calculations and considering multiple scenarios with prudent parameter estimates, we have the opportunity to present a well-justified estimate of business value.

Planning an External Monitoring Operation

Moving from justifying the business case to planning the rollout of an external monitoring service, we will consider three planning tasks:

- Defining external monitoring key performance indicators (KPIs)
- Identifying geographic regions from which to monitor
- Frequency of monitoring
- Defining types of monitoring tests
- Determining how to share monitoring information with service partners

These are the first steps in determining specific details for your business' use of external monitoring.

Defining External Monitoring KPIs

For our purposes, we group KPIs into technology- or business-oriented categories. Some of the most important technology KPIs include:

- Service availability as measured by the ability to complete a standard transaction in a defined period of time.
- Throughput, or the number of standardized transactions that can be completed in a specified period of time
- Statistics on the most popular landing pages
- Average time to complete a transaction

In all of these KPIs, a transaction includes all the steps necessary to complete a piece of business, such as browsing and searching in a catalog, viewing multiple products, filling in Web forms, confirming order information, and so on.

The purpose of technology KPIs is to provide summary metrics that IT professionals can use to readily assess trends in service delivery and platform operations. Comparable KPIs are needed to support business analysis; these can include measures such as:

- Number of transactions completed in a period of time
- Abandonment rate
- Revenue generated by a service
- Average daily cost of operating service

Business KPIs are useful for monitoring the value of services to a business along with their costs. Together with technology KPIs, they provide an indication of the short-term status of services; over time, they indicate trends that may have longer-term impacts on service delivery and profitability.

Identifying Services to Monitor and Frequency of Monitoring

KPIs are metrics that can be applied to a range of services. Another part of the planning process is determining which services to monitor. This will depend on the criticality of the service.

Not all services are created equal. Customer-facing services that support revenue generation are obviously high priorities. If the customer Web site and product catalogs are down, a sales channel is blocked. Customer support services closely follow those services directly related to revenues. For example, if a customer cannot verify the status of an order, that might not immediately affect the ability to generate revenue but might lead to longer-term dissatisfaction with overall service.

Other critical services are those that support core workflows. If vendors cannot check your inventory levels, they cannot run an efficient just-in-time supply service. If Web services you provide to business partners are not deemed reliable, partners will find alternative, probably less-efficient means of getting a job done. In both examples, the ultimate increase in cost will likely be borne by your business.

After defining KPIs and identifying services to monitor, the next planning consideration is how frequently to monitor. The answer to this question will depend on the criticality of the service and the frequency with which it is used. A customer-facing Web site that is used 24 hours a day, 7 days a week will require more frequent monitoring than a reporting service used once a month by business partners. A general rule of thumb for determining intervals between running monitoring tests is that they should occur frequently enough so that you detect performance problems before your customers.

Identifying Geographic Regions to Monitor

The purpose of external monitoring is to understand how your services are performing from the perspective of customers, business partners, and in some cases, remote offices within your business. You should plan to execute test scripts from multiple locations around the globe that reflect the distribution of service users.

In addition to considering business-specific services, remember that changes in global demand for Internet services, especially trans-oceanic services, can have long-term implications on service delivery. For example, according to the research company Telegeography, the existing excess capacity in trans-Atlantic telecommunications cable may be fully utilized by 2014 (Source: Stephen Lawson, "Trans-Atlantic Cables May be Filled by 2014" PC World June 13, 2009. http://www.pcworld.com/businesscenter/article/167143/transatlantic_internet_cables_may_be_filled_by_2014.html). Assuming the most basic law of market economics, the relation of price to supply and demand, this increased scarcity could lead to higher telecommunications costs to provide the same level of service. Long-term external monitoring data can help businesses better understand their telecommunications requirements, which in turn will help better manage costs.

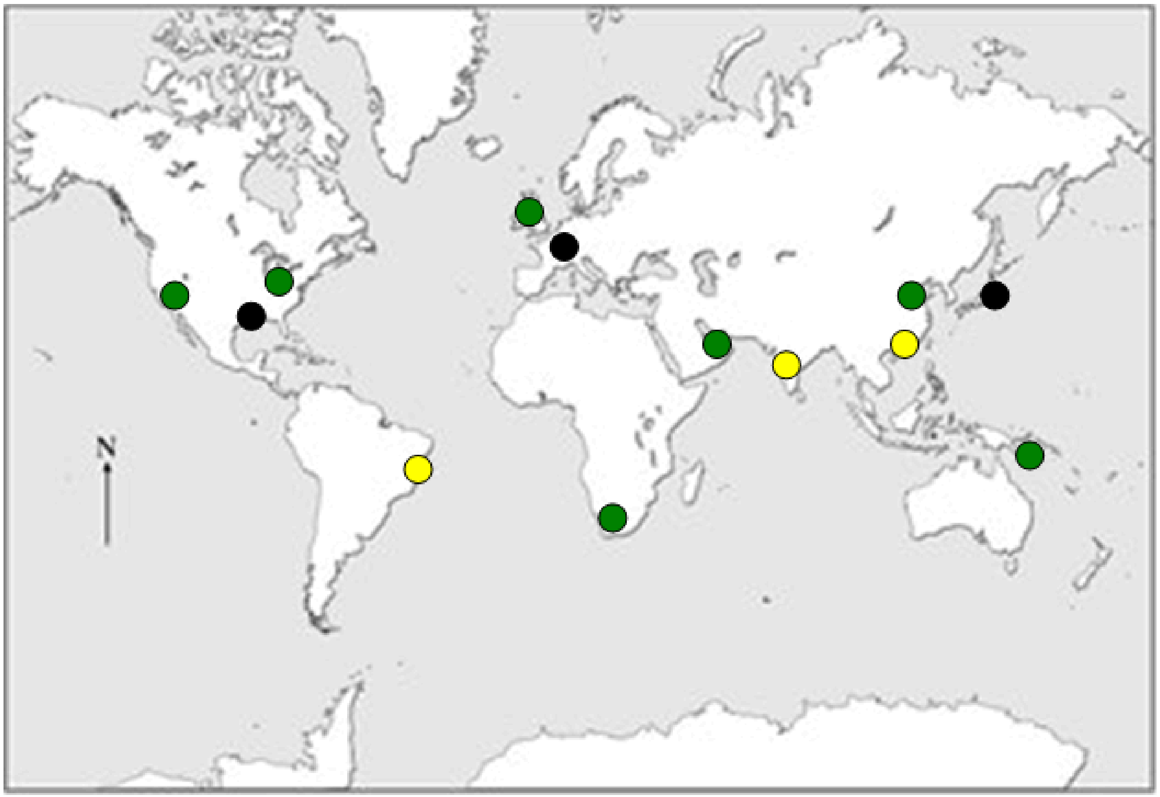

Figure 4.2: Monitoring regions should include sites with high proportion of customers (green), business partners (yellow), and internal users (black).

Defining Types of Monitoring Tests

Different types of services will require different types of tests. Web services and other APIs can be monitored with batch-oriented scripts whereas interactive applications will require browser-based testing. Web services testing is a less complex process than testing rich Internet application interfaces. With Web services, tests should be designed to

- Determine whether the service is available
- The time required to return results
- Verify valid results

Web services testing should include more than simple 'ping' tests that test availability. A range of inputs should be used to exercise the range of functionality in a service. For example, a Web service that calculates the shipping cost and delivery data of product shipments should be tested with different parcel shippers. The top-level Web service might depend on services provided by individual shippers, so the test suite should be sufficient to exercise all major sub-services.

Tests of interactive interfaces should be equally comprehensive. Web applications are increasingly complex. Frameworks such as AJAX and interface platforms such as Adobe Flash enable rich Internet applications that are well beyond simple HTML pages and forms used in the past. Interactive application testing should be browser-based to ensure the tests properly simulate user interactions. Also, test suites should include tests of all core functions offered by the application. For example, an online retailer should test:

- Logging in of customers
- Searching products
- Browsing catalog
- Adding products to a shopping cart
- Checking out
- Looking up order status
- Submitting customer inquiry

Streaming audio and video services, if available, should be tested as well. Streaming media depend on different backend infrastructures, so it is important to test those separately from other backend services, such as databases.

In addition to Web services and applications, basic Web site functionality should be tested. Commonly used pages should be retrieved and timed in test scripts. Also include tests for lower-level network services, such as DNS, FTP, POP3, and SMTP. Problems may manifest themselves earlier in these low-level tests, helping avoid disruption of applications that depend on these lower-level services.

Sharing Information with Service Partners

The valuable information provided by external monitoring may be of use to service partners who need to monitor their own systems, including those that depend on your services. As you would be sharing information with other IT professionals, you have the option of getting fairly detailed in reporting if you like. Summary reports and graphs, such as the examples shown in Figure 4.3, can provide high-level overviews of the state of service performance.

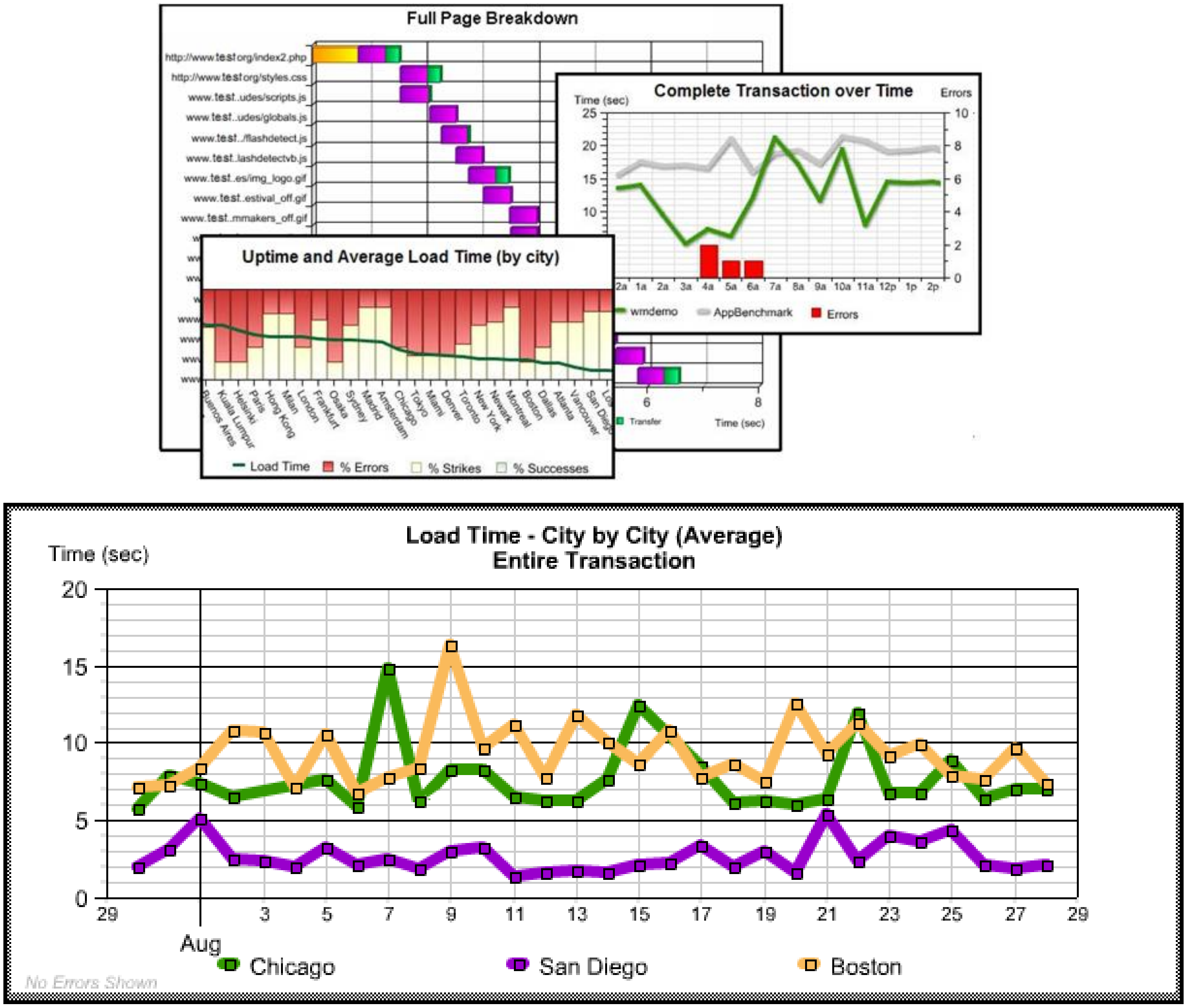

Figure 4.3: Summary reports and graphs can provide a concise overview of the state of service performance.

Planning for external monitoring services should incorporate several considerations, including defining external monitoring KPIs, specifying geographical regions to monitor, defining types of tests to use for monitoring, and finally specifying the type of information to share with service partners. With well-defined requirements specified for these issues, one can move on to the design and implementation phase.

Design and Implementation of External Monitoring

External monitoring will, in most cases, be a service-based solution. Maintaining testing agents around the globe is not a reasonable option for most businesses. Service-based solutions offer many advantages over the "do it yourself" approach, including lower initial infrastructure costs, low telecommunications costs, and lower maintenance. Common drawbacks of service-based solutions—lack of control over implementation details and lack of flexibility when it comes to customization—are less of a problem in monitoring so long as scheduling, reporting, and test scripting are robust enough for your needs. There are still, however, a number of implementation issues that need to be attended to:

- Developing robust test scripts
- Ensuring robust browser-based testing
- Deploying external monitoring scripts
- Integrating external monitoring reports and alerts with existing workflows

As noted earlier, test scripts should assess critical functions thoroughly. This requires testing with a range of inputs that will check multiple execution paths through an application. It is not practical to try to test all features. We are primarily concerned with performance of a representative sample of services, Web pages, and network services. We should remember that performance testing is not the same as software development testing where we run large sets of tests trying to execute all features in a program. Presumably, production code has already been tested in that way. The purpose of external monitoring testing is to ensure that applications are running as expected; testing a subset of services is sufficient for that.

When developing test scripts for interactive applications, it is best to use a browser-based testing framework. These frameworks allow test scripts to test applications the way users work with them. Watir, short for Web Application Testing in Ruby, is one such framework (http://watir.com/). In this freely available, open source application, test scripts are written in the Ruby scripting language, but it can be used with Web applications developed in any language and used with most of the major browsers. One of the advantages of developing scripts using a scripting language is that it is often easy (relatively speaking) to re-use code across scripts. Once a login testing script is written, it can be used as a building block for any number of other scripts.

Scripts should be deployed to agents in a methodical manner. Scripts may be sent to different agents, so it is important to have a method for managing which scripts are deployed to which agents. If all service partners are either in North America or Europe, there is no point in deploying test scripts for service partners to agents in Asia or South America. As with any software, scripts will require maintenance. As new versions of applications are rolled out, external monitoring scripts might need to be updated to test additional functionality.

After the scripts are deployed, they will generate performance data. This data should be incorporated into existing monitoring workflows to maximize its value. This can include sending alerts to systems and application administrators in response to critical events as well as incorporating KPI data into IT management dashboards and line of business management business intelligence reports.

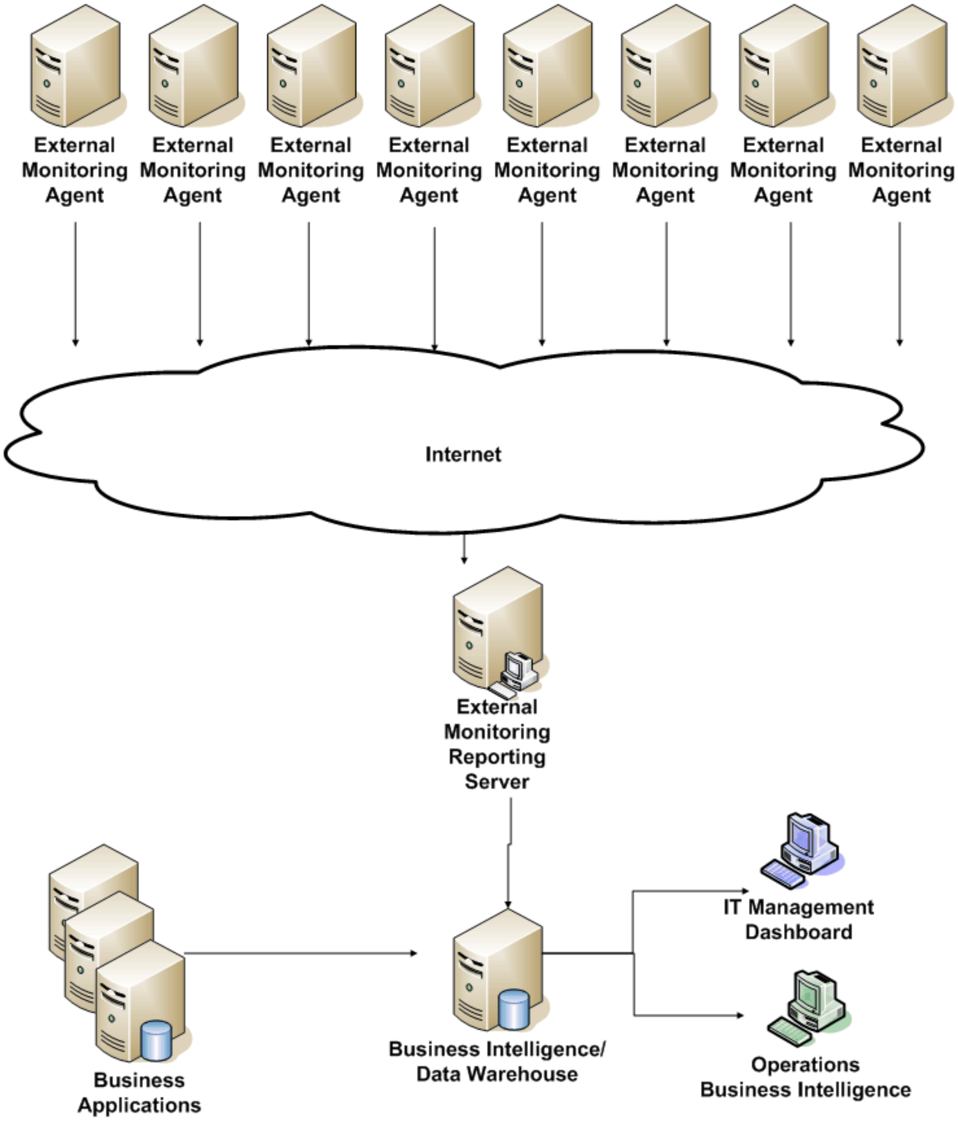
External Monitoring Agent
External Monitoring Agent
External Monitoring Agent
External Monitoring Agent
External Monitoring Agent
External Monitoring Agent
External Monitoring Agent
External Monitoring Agent
Internet
External Monitoring Reporting Server
IT Management Dashboard
Business Applications
Business Intelligence/ Data Warehouse
Operations Business Intelligence

Figure 4.4: External monitoring reporting should be integrated into existing dashboard, management reporting, and business intelligence reporting systems.

Testing and Acceptance

A common part of application deployment is the testing and acceptance process. External monitoring services should be subject to this control as well. The goal of this process is to simulate poor performance and failures so that one can verify a known slowdown or failure is properly detected and reported. Test servers used by software developers are good resources for this type of testing. An example of such a test would include:

- Deploying monitoring scripts to several agents. Each testing agent should also have a human participant at the same location to record response times.
- Running those test scripts several times over regular intervals to test the performance of services running on testing servers
- Collecting monitoring data and verifying that reported performance matches what a human tester experiences from the same location as each test agent.
- After it is determined that the external monitoring system is accurately reporting response times, placing additional load on the test server and run the test scripts. Vary the load so that critical alerts are generated in some cases; in other cases, simply slow the response time of the service and verify the slowdown is properly reported.
- Completely shutting down a service to simulate a hardware failure or fatal software error. Verify the service is reported as unavailable.

These steps can be varied as needed to test more services. Also include all types of services, such as interactive applications, Web services, streaming media services, and network services when conducting acceptance testing.

In addition to the planning, designing, and testing that goes into an external monitoring service deployment, there are management and governance decisions to be made. One of the most important is how a business will communicate performance data with customers.

Communications and Event Handling

Routinely communicating performance information with customers has several advantages over keeping it internally. Sharing data acknowledges to customers that your business recognizes the importance of reliable, responsive services and that your services are monitored. It can also provide customers with information about performance over time so that they can better understand performance peaks and bottlenecks. For example, if there is a regular dip in service response times for a brief period every night due to backups, customers and service partners might adjust their own service use to avoid those times.

Communicating during unexpected performance problems is also valuable for customers. If customers are experiencing a slowdown, the company Web site could display a message about that fact along with estimates of when service response times will return to normal. Even such basic information as that can help customers decide whether to continue to work with the service during the slowdown or return later. Without an acknowledgement, customers are left to make their own uncertain guesses about the performance and reliability of your services.

Governance and Policy Development

Governance and policy development are essential to the long-term viability and utility of external monitoring. Basic policies should be defined during the planning and early deployment phases to specify operational guidelines related to:

- The types of monitoring tests to run and the frequency with which to run those tests
- Reporting procedures
- Alerts and problem escalation

Policies should be defined in such a way that they align with business objectives as well as compliance efforts. External monitoring is not an end in itself; rather it serves some broader business objective related to service delivery. Those same objectives should inform how we formulate policies. For example, if the business is initiating a major program to expand market share in new geographic regions, service delivery to that region should be monitored to mitigate the risk of technical problems undermining a sound business strategy.

Compliance efforts may also benefit from external monitoring. The additional data provided by external monitoring can complement security and audit controls by demonstrating the ability to detect anomalous performance issues, such as large, unauthorized data transfers from a production server.

Long-Term Maintenance of External Monitoring Services

External monitoring services that are provided by third parties will have some, but fortunately not all, of the long-term maintenance requirements of enterprise applications. For the most part, maintenance issues will tend to be related to:

- Maintaining monitoring scripts so that they test new features of services
- Adding new scripts to test new services
- Retiring scripts for services that have been decommissioned
- Adding agents as the base of service users expands into new geographic areas
- Adjusting the frequency of tests according to demands for a particular service

External monitoring is itself a service, so SLAs should be in place specifying the levels of performance your business expects from external monitoring providers.

Summary

There are demonstrable technical and business needs driving the adoption of external monitoring. Businesses can efficiently and effectively deploy an external monitoring service if they follow a logical and methodical process that begins with justifying the business case for external monitoring. IT and business professionals should carefully plan for the types of KPIs that would best serve the business, determine where monitoring agents should be deployed, and determine the types of tests that should be run. Design and implementation phases should be followed by testing and acceptance phases to ensure business requirements are met. Finally, governance and long-term maintenance issues should be assessed soon after deployment to ensure external monitoring continues to provide effective and efficient performance monitoring services.