

Roadmap for Effective Performance Monitoring at a Reduced Cost

Objective

The continued evolution of web strategies and Internet technologies require IT executives to reassess the decision-making criteria for web application delivery in order to adapt and leverage the latest changes in market conditions. This document provides a roadmap for web delivery teams to reduce operational expenditures, and shows how to improve the effectiveness of critical information required for decision-making for Internet services.



Recent Changes in Today's Availability and Performance Requirements

Higher Performance Expectations

Expectations for web application managers have never been higher or more exacting. End users expect services to be well-designed and easy to use. The expectations for performance and availability have ratcheted up to meet the “always-on” demands of a global marketplace. What previously would have been considered “best practice” for measurements (e.g., web page load times of eight seconds or less) is increasingly considered average performance.

Technical and Commercial Trends

Current technical and commercial trends, coupled with new concepts and delivery models, have created the perception within the software application market and software development community that all architectures are nearly infinitely scalable – and that additional capacity can be acquired on an “as-needed” basis at low incremental cost (or even free) without compromising existing service delivery. Some of the factors that have precipitated these expectations are Cloud Computing, Software as a Service (SaaS), falling hardware costs, open source software development and high connectivity/bandwidth.

New Financial Constraints

Given the current economic climate, the most pressing concern for web application and delivery managers is budget expenditure and return on investment (ROI). These market conditions have mandated that every invoice and expenditure is put “under the microscope” to determine its necessity, its contribution to the service providers’ offerings, and how its absence would result in greater risk or cost to business operations. This has resulted in severe strain on the existing requirements/resource balance. Many traditional processes in IT Development, QA, and Operations business functions have come under exacting scrutiny, requiring stringent review of existing practices.

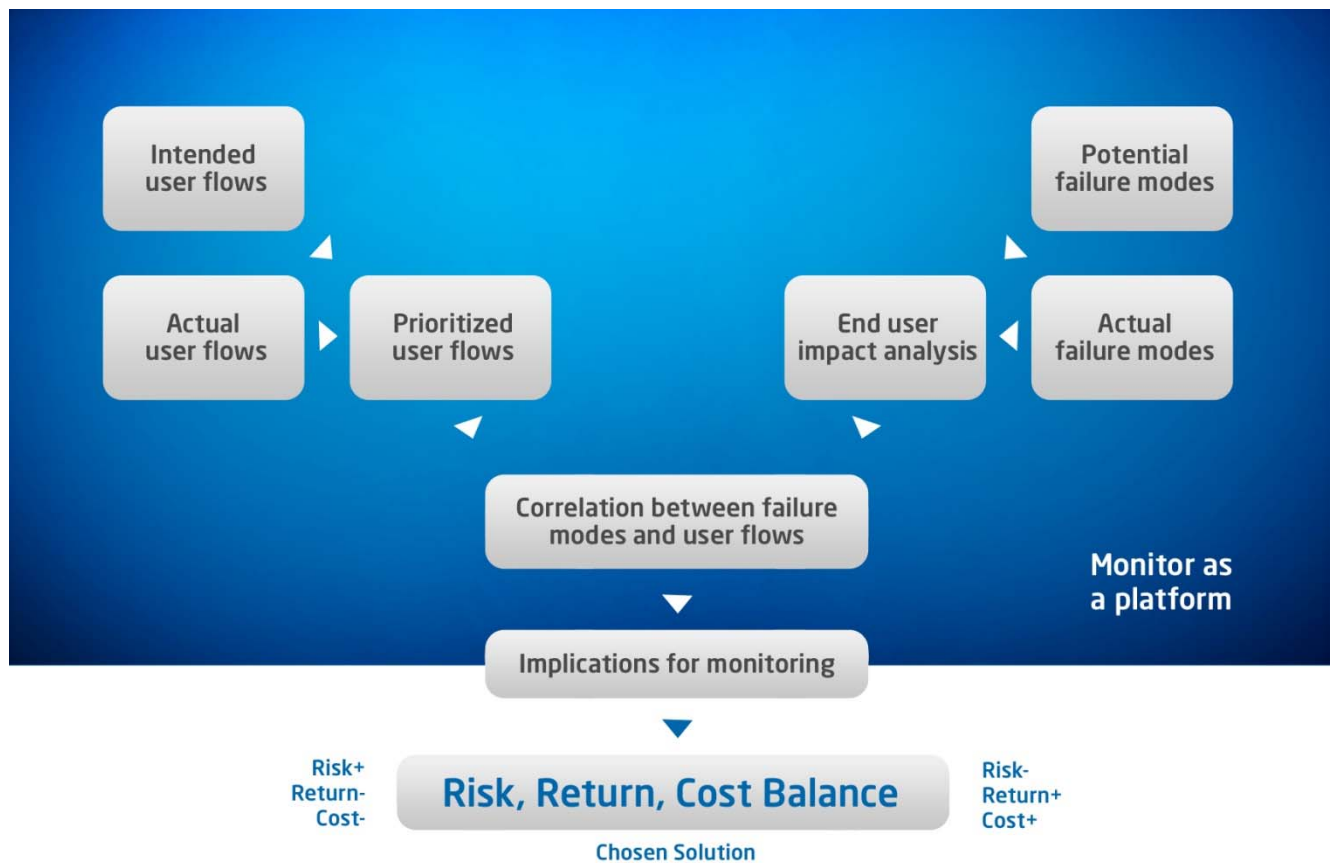
For those concerned with service availability and delivering high-performance web applications, the current economic climate mandates a complete reassessment of how external performance monitoring is used and how this information is interpreted by web delivery teams and transmitted across other business functions within and outside the organization. It is critical that IT operational teams analyze their changed environment and determine how to adjust their existing monitoring strategies to more effectively achieve their enhanced objectives.

More Effective Monitoring at a Lower Cost

There are a wide variety of performance monitoring strategies. This paper identifies some best practices for more effective and cost-efficient performance management.

There are two principal recommendations for monitoring strategy:

- A conceptual approach to monitoring: monitor delivered web applications as a platform
- A user-centric approach to failure analysis



Approach Monitoring Intelligently: Monitor Your Delivered Web Applications as a Platform

Problems with Traditional Approaches

Traditional approaches to performance monitoring adopted the mantra of “monitor everything that can fail.” While this is indeed comprehensive, it has three significant drawbacks:

Cost

Today’s applications have become more complex, both programmatically and systemically – but with the performance management budget under severe constraint, monitoring has become increasingly less effective and more inconsistent. For example, the latest feature enhancement is monitored in preference to some basic (but fundamental) functionality, leaving critical holes in the reliability of delivered services and an information deficiency about their availability and performance.

Information Overload

Conversely, when failure does occur, it is frequently apparent that key members of staff become overloaded with notifications and alerts. Unfortunately, the information is frequently symptomatic of the problem rather than indicative of the root cause. Each alert clutters the recovery process, disrupting the resolution (as it is ineffective in pinpointing the cause quickly), ultimately increasing Mean Time to Failure (MTTF), and damaging uptime statistics. This scenario is particularly likely if performance management is deeply embedded in the organization or if the same provider has been used for some time and the monitoring strategy has gone unchanged.

Lacks Scalability

A traditional approach monitors every point of failure from every customer’s perspective. With increases in functionality, complexity and number of customers, this becomes geometrically expensive, as well as extremely complex to manage and deploy.

Recommendations

Monitoring every end point or every web page, although authoritative, can be expensive and complex to execute. Service providers should focus on functionality that is indicative of the health of their *platform*. The underlying assumption is that if key aspects of the web application are fully functional, then by extrapolation, all aspects of the application are working. Clearly, care needs to be applied with this interpretation. The steps outlined in this white paper cover these in depth.

Reduction in Monitoring Costs

Webmetrics was asked to assist a managed applications service provider with their performance management. Their traditional approach mandated that every service provided to each customer be monitored in order to demonstrate their QoS requirements. The result was the monitoring of a large number of transactions, with several key transactions per customer – but each transaction only lightly covered the delivered functionality.

The Webmetrics platform approach indicated a more cost-effective approach to monitoring. As the service delivered was fundamentally delivered on a flexible and extensible system similar to a Content Management System (CMS), a number of core transactions indicated the health of the system. By augmenting these core transactions, a significant number of internal and external dependencies were validated – providing greater coverage at significantly reduced costs.

Comparison of End Point Monitoring Versus “Platform-Centric” Approach

Traditional Approach	Platform-Centric Approach
FOCUS	
<ul style="list-style-type: none"> • Monitor every failure point • From every customer’s perspective 	<ul style="list-style-type: none"> • Monitor the core or platform transactions as uniquely as possible and extrapolate to as many scenarios / customer perspectives as possible.
IMPACT	
<ul style="list-style-type: none"> • Expensive • Prone to information overload • Not scalable 	<ul style="list-style-type: none"> • Cost-effective • Monitoring information is focused on problem detection and recovery • Scalable to additional functionality, additional complexity and additional volume / customers

A User-Centric Approach to Failure Analysis

Determine User Flows from the Outside In

Service teams are encouraged to look at their end user flows in two ways and to correlate this information:

- User flows desired by the service provider
- Actual user flows from historical usage patterns

Ideally, this analysis should be done with the participation of multiple stakeholders:

Business Function	Contribution
IT Operations	responsible for availability and performance
Product Management	experts in user behavior and knowledgeable about future product roadmaps
Product Marketing	experts in product and service messaging
Customer Service	experts in user problems, work-arounds and resolutions
Technical Support / Second line support	responsible for technical troubleshooting
Marketing	responsible for site design and layout

Example of User Flow

(sometimes called “user journey”)

1. Go to www.store.com
2. Search for “Shirt”
3. Click on “Kid’s Logo T-Shirt”
4. Chose a size, color, then click on “Add to Bag”
5. Click on “Check Out”
6. Fill out billing information
7. Click on “Ship to this Address”
8. Click on “Continue Checkout”
9. Fill out credit card information
10. Click on “Purchase;” verify an order number is returned

Intended User Flows

Service providers should also define their user flows in terms of text entry and mouse clicks, as determined by their vision of the ideal user. Each user flow should be determined with a single objective in mind and should be as straightforward as possible (e.g. “Product Purchase,” “View Product Details,” and “Update User Profile” would be three separate user flows). Advanced functionality should be avoided; for example, complex user flows that necessitate 25 user steps are unrealistic and impractical.

Actual User Flows

Assuming that your web application has deployed web analytics (the measurement, collection, analysis and reporting of Internet data for the purposes of understanding and optimizing web usage¹), then service providers can determine their most common user flows or journeys from the use of their existing services.

- First, determine the most common landing pages.
- Second, determine the most popular routes from these landing pages.

¹ Source: Wikipedia

Correlate Desired and Actual User Flows

Next, these two lists of user flows should be *correlated together*. The best way to achieve this is to map the actual user flows onto the intended flows, providing a list of simple user flows ranked by relative traffic. (The results may be surprising and not without controversy, particularly if this is the first time that this has been done.)

Finally, these user flows should be prioritized into three categories:

Category	Description
Essential	Those user flows that all stakeholders agree are critical indicators of the health of the service (e.g., "Performance of the Homepage," "Log into a customer's account").
Important	Those user flows that exercise particular areas of concern. Different business functions have different areas of concern, so these will vary by department (e.g., Marketing will be interested in the performance of the landing page for a campaign. Product Management and Development will be interested in the user flows that require the most complex programming.)
Useful	These user flows tend to be a "catch-all" for all user functionality.

To avoid endless debate about the value of the user flows between the functional stakeholders, it is important to continually refer to the actual or historical user flows and apply the Pareto Principle (commonly called the "80/20 Rule" – reference at http://en.wikipedia.org/wiki/Pareto_principle) focusing on the majority of your traffic, rather than a small number of use cases that each individually are not frequently exercised.

Identify Common Failure Modes

The next piece of essential information is to determine failure modes. This comes from two sources: actual failure modes and potential failure modes.

Actual failure modes

Recent outages or customer support issues tied to technical failure should be analyzed from the end user's perspective to determine how and in what ways the failure would have appeared to the user. For example, "If the database was working incorrectly, the user would receive a time-out error at a step 3 in a user flow 7".

This process is best facilitated by Product Management using information provided by the following teams:

- Customer Service team or by analyzing recent customer support tickets
- Operations and/or Technical Support for analyzing failures that customers didn't report

Common Failure Modes

- Service not available
- Service responding slowly
- Service responding erroneously

Potential Failure Modes

This group should also analyze their technology delivery infrastructure and brainstorm potential failure scenarios and their impact on end customers (hint: some of this information may be contained in the Disaster Recovery Plan).

It is essential that this failure analysis include all technology infrastructure ultimately exercised by an end user – particularly third-party services that may be outside the direct control of the product development teams.

Analysis of these failure scenarios of third-party services is frequently not executed, which can result in a distorted perspective of the end users' experience. This occurs frequently in engineering-driven (rather than market-driven or customer-driven) software companies, and exposes the service provider to significant risk for the reasons summarized here:

As participants increasingly rely on other players to deliver value, their business practices have become intertwined and interdependent. When services provided by one partner slow down – or worse, become unavailable – the effect radiates out to each dependent participant, which may in turn critically impact other internal or external interfaces or systems.

Determine Customer Impact of Each Failure

For the potential failure modes, determine the likelihood of the failure and how the end user would experience the failure. Then, determine how critical this failure would be to the end users' experiences or the potential damage to the service's brand.

Examples of Subject Areas for Failure Analysis:

- Failure of datacenter
- Failure of internal network
- Failure of inconsistent routing (to your site)
- Failure of step in your transaction (e.g., validation of customer's account history)
- Failure of third party web service (e.g., Ad server, cloud server)
- Failure of hardware (e.g., hard drives)
- Failure of client-side processing (e.g., JavaScript)
- Failure of transmission (e.g., Security protocols)

Examples of Commonly Used Third Party Services

- Multiple data centers
- Content Delivery Networks
- Payment providers
- Web analytics
- Internal services (particularly relevant to Service Oriented Architecture (SOA) infrastructures that are used to deliver service to the end customer.

Note: It is important to focus on those services that are responsible for the delivery of web applications, and to exclude services used in the construction and deployment of the web applications.

Issues within an Ecosystem

- Identification is difficult
- Sharing information is difficult
- Symptoms compound root cause
- Correlation is key
- Multiple solutions are required
- Resolution may not work without collaboration
- Partners are likely to participate in multiple ecosystems

For more information, please see www.webmetrics.com/solutions/ecosystems.html.

Example Worksheet

Failure	Failure Type	Customer Impact	Historical Reliability / Probability of Failure	Impact
Failure of Data Center	Service not available	User would receive 404 / Not Found error for any web page	Moderately likely. (Historical uptime is 99.9%)	Very detrimental
	Service responding slowly	Load time of any webpage would respond slowly. Some pages may time out.	Most likely	Highly detrimental
	Service Responding erroneously	Service would receive error message	Unlikely – such problems are more likely to be routing or caching problems	Highly detrimental
Failure of Third Party Ad Service	Service not available for some users	Some parts of some pages would return 404, but service would be functional. Lost revenue.	Estimated reliability 99.8%	Moderate: Lost revenue. Some brand damage as it might confuse some users.
	Service responding slowly	Service would appear to work, but user would perceive 'gaps' in pages	Much more likely, particularly in some regions of the world	Less impactful to user experience, but some customer confusion leading to brand damage
	Service Responding erroneously	Much more difficult to assess. Ranges from not impact to moderately impactful (e.g., totally inappropriate ads or very large ads)	Least likely to happen	Size of served ad can be checked by external monitoring. Incorrect ad would be difficult to assess without good reporting from third-party Ad service.

Correlate Failure Paths with User Flow Analysis

Ultimately, the most essential step is the correlation between failure paths and the user flow analysis:

- Failure analysis identifies the numerous ways in which the delivered service could fail
- User flow analysis reveals the most valuable or commonly trafficked routes

It should be anticipated that many of the failure routes will be trapped by the monitoring of the top priority routes from the user flow analysis. A small number of user flows will also cover many, many failure routes.

Some failure routes will not be covered, however, and these need to be reinvestigated with care as to the probability of occurrence and whether they are covered, either entirely or in part by other user flows. This step represents an important validation of some of the assumptions made to date. Interestingly, some use cases that occur very infrequently are uniquely critical – usually testing a vital piece of functionality that represents a bottleneck or a single point of failure.

Finally, it is worthwhile ensuring that sound and proper development, QA and release processes are being executed, as trapping issues in the development QA cycle is much more effective and cost efficient than simulating real users using external performance monitoring. On occasion, some highly proficient operational teams take on responsibilities that, in reality, should belong to others; weak or inconsistent release processes are the most likely culprits.

This useful step validates much of the user scenarios that operational teams use traditional monitoring to uncover. It also exposes some “holes” in the existing monitoring strategy – most usually in the measurement of third-party services.

Identify External Monitoring Requirements

The list generated from the previous step represents the ideal or comprehensive monitoring requirements.

Risk, Return and Cost Balance

These comprehensive monitoring requirements should now be ordered by several factors:

- Value/Utility of service being monitored
- Cost of customer impact (factored by likelihood of occurrence)
- Cost of external monitoring
- Cost of resolving identified problem (e.g., the failure of a data center can be very time consuming to resolve, and hence numerous mechanisms exist to mitigate this risk, including backup and disaster recovery plans, data replication, multiple data centers, geo-balancing, etc.)
- Actionable data from monitoring (see side panel for explanation)

At some point, the return on monitoring investment starts to diminish to unacceptably low levels. This point will be determined by factors, the most important being budget.

Cost Saving Tip – Redundancy in Monitoring Data

When choosing monitoring services, being mindful of the actionable data that monitoring provides can save both time and money. Acquiring performance information can become an expensive science if the information is only “useful” rather than “actionable”. For example, knowing at exactly what step a user transaction or journey fails is actionable by the service provider’s team. Technical support can trace logs back to around or before that failure and work out the dependency. However, information on individual ISP performance is less actionable and can lead to additional costs. Ultimately, your customers are routed to the internet through the best possible path, utilizing numerous ISPs. Your monitoring platform can be more effective if it reflects this end-user model. Monitoring performance through the view of a single ISP causes inactionable alerts by notifying you when a single ISP is affected – an “error” that you don’t have control over. In addition, if you want to get a true view of end user performance, you would need to monitor through multiple ISPs, leading to additional costs.

By executing the series of steps outlined in this document, service providers will be able to precisely identify the value of monitoring data with respect to their service.

Conclusion

In financially difficult times, ROI is carefully scrutinized. Using the conceptual approach and the practical steps outlined in this white paper, service providers will be able to gain much more effective use of their performance monitoring information – and do so very efficiently.

By using the series of steps outlined, additional benefits are likely to be gained. Service providers will be able to more effectively triage their service in the event of operational failure. Furthermore, they will be well-positioned to update their customer service procedures, technical support resolution guidelines and disaster recovery plans with respect to service failure, poor performance or intermittent problems.

Finally, by bringing all the stakeholders into the analytical process, awareness and understanding will be generated – which, in turn, may initiate a review in development, QA and release processes.

About Neustar

Neustar (NYSE: NSR) provides market-leading and innovative solutions and directory services that enable trusted communication across networks, applications and enterprises around the world. Neustar Webmetrics services provide detailed availability and performance analysis, allowing for better customer-centric decision making. Webmetrics analysis provides visibility into business and technical interdependencies across Web ecosystems and clouds. Capabilities include:

- Ecosystem Management is a monitoring and collaboration system that allows for effective management of third-party services relied on in the cloud and provides secure tools for sharing performance information with internal and external customers and partners.
- Application Transaction Monitoring simulates defined Web transactions, such as purchase order fulfillments, or customer logins, to provide detailed information on their performance from an end user perspective.
- Website Monitoring provides global awareness of Website availability with breakdowns on areas such as DNS time, time-to-first-byte, and transfer time.
- Web Services Monitoring monitors REST/SOAP Web service requests via XML request/response and content verification with additional support for transactional capabilities.
- Stream Media Monitoring allows for customizing the length of time to watch a stream and setting thresholds for connection timeout, buffer timeout, stream quality and other metrics.
- Network Services Monitoring monitors DNS, FTP, Ping, POP, and SMTP protocols to assess availability and connectivity.
- Load Testing services provide real-world validation that web applications are ready to go, with a fully managed solution that saves time and money.

For further information about Webmetrics products and services, call 1-877-524-8299 or email sales@webmetrics.com. Visit us online at www.neustar.biz or www.webmetrics.com.