End-user Experience Monitoring

Nimsoft Whitepaper

End-user Experience Monitoring

Table of Contents

Nimsoft Whitepaper

End-user Experience Monitoring

Executive Summary

According to Forrester Research, “end-user experience has to be the benchmark for enterprise application performance[1],” which of course makes perfect sense, given that enterprises are spending billions of dollars to enable productivity and automation gains as specifically directed by end-users. These processes are critical to the bottom line operation of a business and in most cases end-user performance degradation can be directly correlated to a negative business impact.

While the need for end-user monitoring (EUM) might seem obvious, the reality is that a majority of companies are just now embracing this capability as part of their overall IT service management strategy. Furthermore, those companies that have already invested in end user monitoring solutions are discovering shortcomings that limit the effectiveness of their contribution. It is critical that EUM metrics be incorporated into a complete end to end view of your IT hierarchy, including direct integration with the wealth of performance data associated with legacy infrastructure monitoring tools.

This integration provides valuable supporting data that in turn further leverages the value of the true critical perspective, the end-user! This paper will highlight the overall benefits associated with end-user monitoring and also outline the various techniques available for capturing this critical perspective. We will also explore the role of EUM as a key component of an IT Service management strategy that will deliver significant improvements in efficiency and effectiveness when compared to traditional IT management solutions.

What Exactly is End-user Experience Monitoring

The term refers to methods and processes used to systematically assess systems performance from the user’s perspective. The techniques encompass two types of approaches:

- Active (also termed synthetic) monitoring which uses scripted transactions to measure performance and availability. We will refer to this as End-user Experience Monitoring (EUEM), and
- Passive monitoring of actual user interactions with the system, which we refer to as Real User Monitoring (RUM).

Why End-user Experience is Important to the Business

IT organizations are increasingly being required to support applications that serve the needs of the business’ customers, employees and partners. The experience of end-users is especially important for web-enabled, customer facing applications because competition is often only a click away. Converting more visitors into customers can make a huge difference on the bottom line in today’s friction-free world of e-commerce.

End-user experience is also important for internal users because of the high cost of discontent. As described previously, by their nature, enterprise applications and systems mostly have the risk of being dissatisfiers. In most cases it would be fruitless to attempt to measure end-user delight; we can only measure end-user dissatisfaction (actually, only the causes of dissatisfaction, e.g.: usability issues, network delays, sluggish servers, or transaction errors; or the results of it, such as the number of support calls or trouble tickets). Disgruntled users are often too busy to report every issue they encounter, but will anecdotally complain to others

[1] Jean-Pierre Garbani, Forrester Research 2006.

about the application. Eventually user complaints will ripple up though the management hierarchy and may blindside the CIO. At that point the gap between traditional operations tools (“Everything is green and our monthly scorecards show full goal attainment”) and a much lower end user satisfaction level start to discredit the IT organization. A reactive fire-drill may also occur, creating additional workload for IT support staff and end-users, further exacerbating the problem.

We are measuring dissatisfaction in terms of performance outside the range of established thresholds and expectations. We are identifying (not measuring) the causes of dissatisfaction.

When the end-user community is content, IT staff is free to focus on planned activities that bring incremental value to the business (rather than responding to interruptions which attempt to minimize loss of revenues, productivity and profits). IT will have a difficult time becoming a strategic partner to the business if it always operates in tactical mode of fighting fires or the crisis du jour.

- Nearly 85% reported experiencing incidents of significant application performance degradation
- Over the last 12 months, 51% acknowledged instances of poor application performance growing more frequent
- Incidents had at least moderate impact on employee productivity (82%), team productivity (77%) and customer service quality (79%)
- More than 50% admitted launch delays of new applications because of network performance concerns

—Forrester Research, April 2004

Measuring the real end-user experience offers benefits in a number of areas. IT gains a better understanding of application usage patterns before taking an application live. Verification of the impact of configuration and tuning parameter changes can be immediately available. Researching a problem isolated to a specific user becomes more practical. Baselines can be established and used as benchmarks to quickly recognize a problem such as a slowdown or outage. The results of real end-user monitoring also point the way towards areas where additional investments would be justified to improve user productivity.

Improving end-user experience has two discrete phases: Understanding what users are actually experiencing and making necessary improvements. The speed of resolving user dissatisfaction issues is the critical factor in increasing user satisfaction. For example, Dell has discovered that speedy resolution of customer issues builds customer loyalty—in an industry where product differentiation is difficult. Rapid isolation of the root cause is the first step of problem resolution: is this a server problem, a network issue, or a condition limited to the desktop?

The next step is to compare the measured results to historical benchmarks. The management solution of choice should record and aggregate historical data for analysis. Current measurements, when compared to such historical norms, offer strong clues to the nature of the problem. Examples:

- Is the performance glitch caused by an upgrade to a new version of a web server, which inadvertently was misconfigured in a manner that significantly impacted performance for WAN users?
- How did performance change when we converted another user location to the new application and what performance trends can we predict as we plan on adding additional workload when we convert our largest site next weekend?
- Are users sporadically getting problems when they are routed to a particular load-balanced server in the pool?
- Are there any differences based on geographic location, suggesting a network-based problem?
- Are users getting cryptic error messages (“Could not complete transaction—error 7432 ”, “Microsoft OLE DB Provider for ODBC Drivers error ‘80004005’” or “Exception System. Out Of Memory Exception was thrown”)?

We find that IT operations staff often is being asked to support more applications and users without headcount growth. IT typically does not have the luxury of time to contemplate the symptoms and potential responses; solutions must be applied promptly to avoid costly impact to the business.

The Subjectivity of Experience

The experience of end-users of enterprise software is subjective and quantifying it has proven elusive. There are several reasons for this.

- Performance. Enterprise application performance is not a satisfier; it can only be a dissatisfier. No one shouts with joy from the rooftops when an SAP transaction completes successfully for that is the expected norm. It is easy to see, on the other hand, when a person is frustrated by not being able to accomplish a required task in a timely manner—and they will complain to others about it!
- Consistency. End users often grow accustomed to a given level of performance. When performance is erratic and unpredictable they get frustrated.
- Usability. Behavioral studies of end user interactions suggest that end-user experience depends on much more than screen layout design and application interfaces. It might be useful to think of the "AURA" of user experience depending on...
- Availability—Is the application or system ready to do business? End-users increasingly have an expectation of a utility-like, or dial-tone (99.999%) availability.
- Usability—Is the application easy or difficult or cumbersome to use? This is where complexity, human factors, native language and culture come into play.
- Responsiveness—What kinds of time delays are experienced during routine tasks? What level of interactivity does it offer and does the application keep the user aware during long processing cycles? How predictable and consistent is application performance? What expectation should the user have for the overall response time?
- Accuracy—Does the application complete the entire task without errors?
- To the extent they think about it at all, developers are accustomed to thinking about transaction performance as measured from within the application running on a server, or "inside out". Yet end-users work with the system from "outside in". As a result, server-side performance data often misrepresents the users' true experience. The dichotomy of the "inside out" vs. "outside in" views highlights the ineffectiveness of solely monitoring infrastructure performance. How often are all the server lights green, yet the users are screaming in pain? For example, an application might be instrumented to measure the number of milliseconds of server time to complete a database update (almost always irrelevant) or the total transaction turnaround time (but without any visibility into the additional time required on the network and client system.)
- Business transaction processing almost always entails multiple user interactions with the system. The overall time to complete a process will be influenced by long think times caused by any slowdown that interrupts the momentum and flow a user has with the system. Users rarely will take the time to report on sporadic system performance problems, leaving IT with a blind spot for the true level of user satisfaction.
- Other factors such as training, workspace environment including interruptions and distractions that impair productivity and helpdesk support levels also contribute to user satisfaction; however we will not consider those factors in this paper. In the typical case where the overall business transaction requires multiple user interactions with the system, the overall time to complete a process might also be influenced by long think times caused by interruptions such as phone calls, bosses, instant messages, coffee breaks, and alike. The dichotomy of the "inside out" vs. "outside in" views highlights the ineffectiveness of solely monitoring infrastructure performance. How often are all the server lights green yet the users are screaming in pain?

The richness and variation of interfaces in user applications (traditional green-screen host applications, Windows applications, 4GL-based or ERP applications with their own unique look and feel, Java-based applets and applications, Flash, DHTML/AJAX) make it even more challenging to define a uniform standard of end-user experience.

Measuring End-user Experience

Effective measurement of the end-user experience requires an understanding of the components from which more complex tasks are built. Four different task spans can be measured. In the order of granularity, they are:

- User Events (interaction with a Windows or web form control such as a list box or calendar control).
- Application Steps (entering a line item of a customer order).
- Business Transactions (entering a complete customer order).
- Workflow Tasks (multiple departments collaborating on, for example, booking an order, checking customer credit, allocating inventory, scheduling shipment, and notifying the customer in order to process an order).

> *Commonly experienced transaction problems included error messages (40%), a poorly navigable web site (37%), the inability to complete a transaction due to an endless loop (31%) and difficulty logging onto the website (31%).*
>
> —Forrester Research, April 2004

Measurements can also be classified by where they are taken:

- At the desktop, laptop, or mobile device—where user experience happens.
- On the network—where latency and congestion happens.
- At the server—where resource contention may occur.

End-user experience is the consequence of end-to-end application performance and that of the underlying infrastructure. Granular monitoring serves to understand factors impacting overall performance, aid in root cause analysis and rapid fault isolation.

Meaningful measures cannot be applied to a user who is not using the application properly, for example if the user is confused as to how to properly enter data into a form, or a user that does not know how to use search criteria effectively so they retrieve unnecessarily large amounts of data and spend extra time paging through the results. These sorts of usability and training issues ideally should be resolved during usability testing or acceptance testing phase of the development lifecycle, but frequently are overlooked. If an effective RUM capability is already in place, it can be used during the pre-production user acceptance test period to help pinpoint users who are suffering from reduced throughput based on an application design problem or the users' ability to use the application correctly.

Historically, operational IT infrastructure management tools did not focus on measuring the actual end-user experience. Most of today's monitoring products tend to be point products that piecemeal out the clues to the experience of end users. Indicative believes that end-user experience monitoring delivered with the context needed for true end-to-end service management is going to be required to allow IT operations to rapidly and appropriately respond to performance anomalies. The root cause must be identified and the issues resolved before the problems magnify to impact the business. IT needs comprehensive oversight of actual, measurable variance in end-user experience that is tied to the underlying infrastructure performance—not anecdotal generalities ("the system seems slow") or overwhelming amounts of data pulled from SNMP MIBs by multiple tools. N.B.: The data collected in the course of EUM can also be quite useful in the context of capacity planning.

The Pros and Cons of Various Approaches

The tables that follow summarize the tradeoffs of various approaches to end-user monitoring. Active monitoring relies on robotic agents stationed at key locations in the enterprise to generate application transaction flows. Active monitoring typically requires some degree of script development which is facilitated by a record/playback engine. For straightforward applications this can be accomplished with “point and click” simplicity. Functional testing scripts used in test mode typically need to be substantially rewritten for production monitoring although in some cases there is an opportunity for script logic re-use.

Active (Synthetic or Robotic) GUI Drivers

Pros	Cons
• Support for all application types (web, ActiveX, Java applets, Flash, client/server, SAP GUI, Java applications, Citrix, host emulator, etc.) • Repeatable: exactly the same transaction executes every time. Not subject to variation based on end user machine workload or the varied amount of workload in the transaction • Measures “true” user experience, including client rendering and processing times • Measures total time for multi-step transactions, along with the breakdown of sub-step overhead • Measures across application tiers, to give a true “end to end” view of performance Simple to set up—typically the most popular way IT organizations start to implement end user experience monitoringDatabase cluster monitoring	• Periodic sampling: sporadic errors or errors that only affect certain users may not be detected • Scripting skills are required: scripts must recover from unexpected errors. Scripts also need to be carefully written to handle timing issues that might “hang” a script. • Scripts must be maintained as the application changes • Inherent limitation of the number of scripts that can be run per hour on a single machine (single threaded) • Impractical to monitor actual user experience if a robot is needed behind a user’s external firewall • Robot needs to be “unlocked” and scripts need to have logon passwords to the applications—this can pose a security risk Robot needs to be placed at each key branch or remote user location. • It can be cost prohibitive to do deploy robots at a large number of locations.

Synthetic Protocol Drivers

Pros	Cons
• Supports multiple application types, but typically is used for web applications. • Especially good for availability monitoring of many concurrent transactions. • Transactions can be replayed in parallel (multi-threaded). • Component level times can be captured (e.g.: HTTP page elements). • Load testing tools used by many vendors to accomplish this.	• Hard (some would say near impossible) to build scripts for other than web • Have difficulty handling dynamic application content and client-side scripting (e.g.: JavaScript) • Scripts must be maintained as the application changes • Typically require expertise in a load-testing tool. • Operations usually do not have this expertise • Vendors may tell you that load scripts used in pre-production can be used in production—caveat emptor!

Real User Monitoring Based on Network Appliance (Sniffer)

Pros	Cons
• Excellent approach for external customer-facing HTTP applications where complete end user experience for all transactions is desired. • Correlate page download times with network metrics such as packet retransmissions, TCP connection timing, number of application turns, etc. • Provides component-level timings (elements of HTTP page) • Delineates the approximate network and server components of transaction response time. • Some RUM products can also play back actual user sessions for usability analysis. • Rich data source for performance analysis—e.g. slow users, slow regions, slow times of day, slowest web pages, etc. • Can detect application errors by inspecting the contents of pages being returned to users.Hard (some would say near impossible) to build scripts for other than web	• For SSL based applications the sniffer requires a copy of the SSL Private Certificates (potential security concern). • Transactions of interest must be defined and configured. • Unable to determine actual client rendering times or client side (e.g. Javascript) errors. • May have problems with complex portal-based applications using framesets • Potential security concerns regarding sensitive data • This level of analysis is typically not performed by IT operations. Depending on the vendor and implementation path chosen, a RUM tool may take on more of an episodic project-focused usage as contrasted with a proactive service management focus • Limited to no ability to monitor other than web (HTTP) transactions No visibility into performance of web page content delivered from content caching sources such as Akamai or Cisco Content Engine.

Real User Monitoring with Browser Script (e.g. JavaScript)

Pros	Cons
• Turns each end user's browser into an agent to collect actual performance data. • Actual client rendering time can be captured. • Browser errors, such as incomplete downloads and popup messages, may be collected. • No installation required on the client: this approach works well for trading partner and customer-facing applications.	• Invasive—requires hooks into the web server to instrument the pages. • May need filters to control which pages and/or users are being monitored. • Network load from performance data being streamed from the end user's browser. • Does not support non HTTP based transactions. • Not capable of determining the network component of response time and network-related problems. • May not support complex web applications such as those that use ActiveX controls, Java applets, Flash, AJAX.

Real User Monitoring with Agent on the End User Desktop

Pros	Cons
• Can monitor real user transactions—especially those that update a database and impact a business application. • Capability to monitor local machine resources such as CPU, memory, disk IO. • Potential to capture GUI events such as keystrokes, window frames, and mouse events. • Potential to collect additional data (ARM, browser collected timing data, etc.).	• Requires installation and updates to deploy agents on all end user machines. • Not a good solution for PCs not controlled or owned by the customer. • Can be complicated to define the unique transactions of interest. Often transactions have more than one way to be invoked. • GUIs based on 4GLs or Java (such as SAP and Oracle) may not be supported. • Potential privacy concerns.

Recommended Approaches

The "best" approach obviously depends on your objectives. This paper has outlined the key reasons why IT organizations are implementing EUE monitoring, and the tradeoffs of various approaches to implementing this monitoring. Before you evaluate EUE monitoring products, keep in mind these guidelines:

- Cost of implementing EUE, including the full implementation and ongoing maintenance.
- Scope of EUE—which applications are most critical and the ones where IT is exposed to potentially being blindsided when problems occur?
- Avoiding the "yet another tool" syndrome. Make sure you know how an EUE product will actually be used within your current operations environment.

Usually a hybrid approach is the most practical way to get started, comprised of a combination of synthetic and real user monitoring targeting your most critical applications. It is often a mistake to simply consider end-user monitoring as a new source of event-based data to integrate with existing management tools. A more practical solution would be to frame the data already collected by existing monitoring products in the context of the end-to-end services offered. The end-user monitoring implementation should have at least the following characteristics:

- Measures end-user experience in the context of all underlying aspects of service, such as firewalls, load balancers, web servers, application servers, and database servers.
- Able to provide data in the context of an end-to-end service delivery chain.
- Provides the ability to deliver performance data in a consistent manner for both active and passive monitoring.
- Supports 24 x7 monitoring for all HTTP transactions, from all end user locations—not just sampled data.
- Supports non-web based applications as well as HTTP.
- Is a product you can quickly implement with rapid time-to-value and minimal ongoing administrative costs.
- Provides the ability to help you rapidly get to the root cause of problems.

Planning Guide

Questions to Ask to See if End User Monitoring is for You at this Time

The topic of end-user experience monitoring is both wide and deep. To make sense of it all, we must focus on the most prominent usage scenarios. Here are some questions to help in the process.

- What is the risk of IT getting blindsided by the business units when a problem occurs and IT is not aware of it before the users complain?
- What are the key applications to monitor—based on business criticality, frequency of change, user satisfaction level, frequency of problems, political considerations?
- What are the likely causes of application performance problems?
- Are your current tools adequate to detect the wide variety of performance problems that users might encounter? How pervasive are performance problems?

- Do you have an ITIL initiative in place to systematically prioritize and institute process improvements based on “lessons learned”?
- Can “before and after” snapshots be compared following a tuning change?
- Does Operations have sufficient proof points for developers to implement a change?
- Can these be used as test cases to verify the effectiveness of the solution?
- Is there sufficient correlation (trending) in the data to be useful in anticipating future problems?

About Nimsoft

Nimsoft provides integrated, modern IT management solutions for more than 1000 enterprise and service provider customers globally, including 1&1, CDW, SoftLayer, Sur La Table, TriNet, and Virgin America. The company’s Nimsoft Unified Manager is an industry-leading solution which helps organizations easily monitor and manage IT services in increasingly complex business environments. Nimsoft products integrate with existing solutions at any point from the data center to the cloud. For more information, visit www.nimsoft.com.

North America Headquarters
U.S. toll free:
1 877 SLA MGMT (752
6468) 1 408 796 3400

Email: info@nimsoft.com
Web: www.nimsoft.com

United Kingdom
+44 (0) 845 456 7091

Norway & Northern Europe
+47 22 62 71 60

Germany
+49 (89) 90405-170

Australia
+61 (0)2 9236 7216

Brazil
+5511 5503 6243

Mexico City
+52 (55) 5387 5406

Singapore
+65 64328600

New Delhi
+(91 11) 6656 6667

Mumbai
+(91 22) 66413800