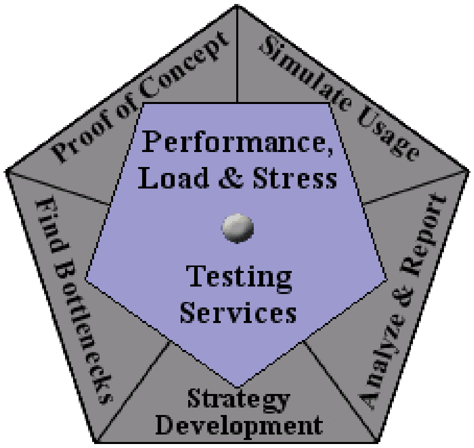
User Experience, Not Metrics

by:

R. Scott Barber



Part 1: Introduction

With so many professionals online and relying on the Internet to perform daily operations, application performance has become vital to the success of an eBusiness solution. In an effort to ensure success, many companies have developed tools and methodologies to test and tune applications for performance. These tools and methodologies have focused around optimizing system metrics, rather than optimizing user experience. The *User Experience, Not Metrics* Series of articles will address topics related to determining true user experience and application performance tuning using Rational Suite TestStudio coupled with a proven approach to engineering end-to-end system performance that focuses on end-user experience.

This introduction is intended to provide the high-level concepts and terminology used throughout the *User Experience, Not Metrics* Series of articles.

Introduction

How many times have you surfed to a website to accomplish a task only to give up and go to a different website because the home page took too long to download? "46% of consumers will leave a preferred site if they experience technical or performance problems." (Juniper Communications) In other words, "If your website is slow, your customers will go!" This is a simple concept that all Internet users are familiar with. When this happens, isn't your first thought always, "Gee, I wonder what the throughput of the web server is?" Well no, that is certainly not the thought that comes to mind. Instead, you think "Man, this is SLOW! I don't have time for this. I'll just find it somewhere else." Now consider this, what if it was YOUR website that people were leaving because of performance?

Face it, users don't care what your throughput, bandwidth or hits per second metrics prove or don't prove, they want a positive user experience. There are a variety of books on the market, which discuss how to engineer maximum performance. There are even more books that focus on making a website intuitive, graphically pleasing and easy to navigate. The benefits of speed are discussed, but how does one truly predict and tune an application for optimized user experience? One must test, first hand, the user experience! There are two ways to accomplish this. One could release a website straight into production, where data could be collected and the system could be tuned, with the great hope that the site doesn't crash or isn't painfully slow. The wise choice, however, would be to simulate actual multi-user activity, tune the application and repeat (until the system is tuned) before placing your site into production. Sounds like a simple choice, but how does one simulate actual multi-user activity accurately?

That is the question this series of articles attempts to answer.

Terms and Concepts

Understanding the terms and concepts below is critical to getting the most out of this series of articles.

Performance Testing: the process of identifying and eliminating application or system performance bottlenecks by exercising actual expected user patterns. See Figure 1.

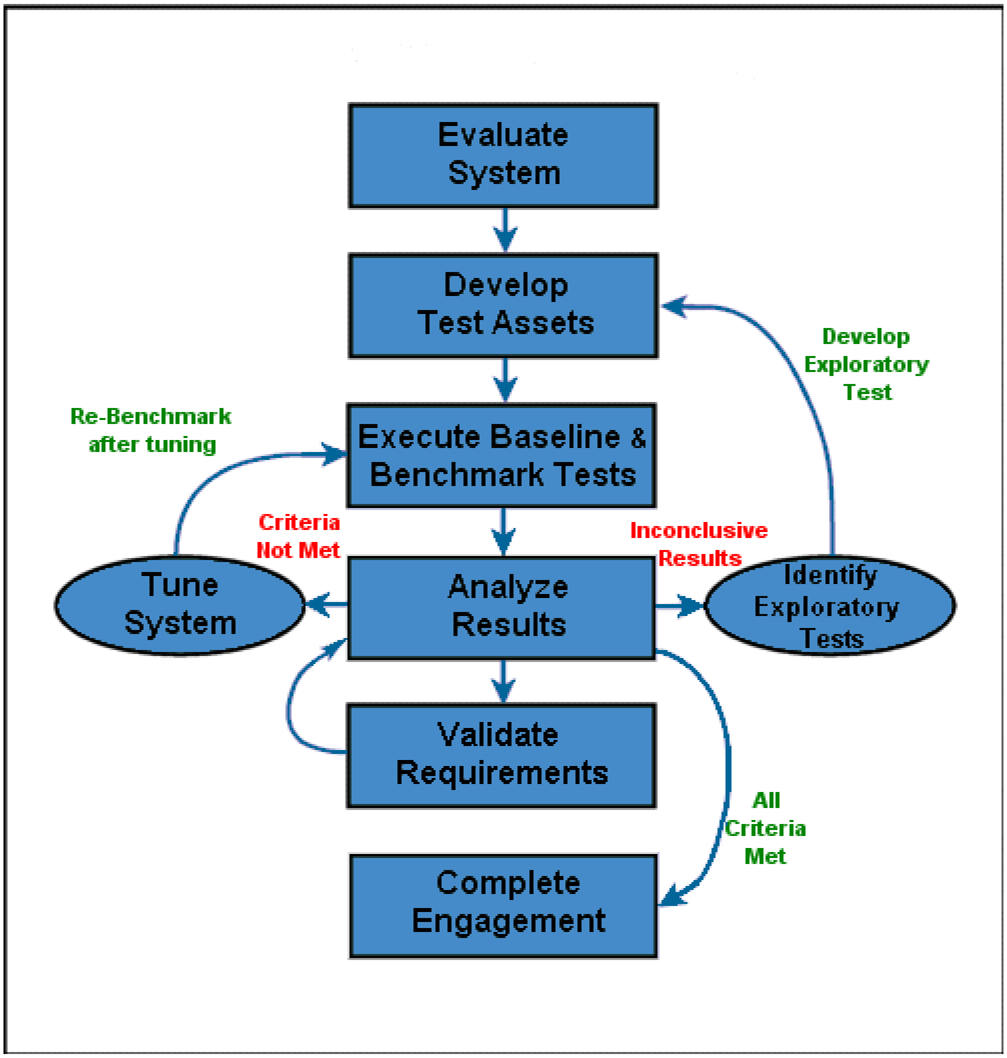


Figure 1: Visualization of Performance Testing

Workload Distribution: a representation of the functions performed by a user community on a system, sometimes known as a User Community Model. For example, during the course of a day on a retail-based website, most users are shopping, some are searching for a specific product, some are finalizing purchases and checking out, while a single administrator may be updating product prices. A workload distribution is based on a percentage of users performing a specific function over a given period of time. Using the above example a workload distribution could be: shopping – 83%, searching

- 5%, checking out – 10% and administration - 2%.

When viewed from a user-experience perspective, **Performance Goals** are expressed as a maximum allowable response time at a pre-determined number of concurrent users.

Response Time is measured from the end-user perspective. For example, the time between when a user presses the “login” button and when the subsequent page is fully loaded.

Session Duration is the total amount of time a single user is using the website during a single site visit (expressed in fractions of an hour). Session durations may vary by type of user.

The number of **Concurrent Users** is the total number of users who are actually accessing the site or who have active sessions at a specific instant in time.

Total Hourly Usage is the anticipated number of users accessing the site in an hour. This is calculated by multiplying the session duration by the number of concurrent users.

Baselines, in this context, are single user, single script tests that are recorded to provide a starting point for time comparisons.

A **User Delay** is a wait time incorporated into a script so that when that script is played back it plays at the same pace as an actual user.

This series of papers may use some common terms that have different meanings to different people. These terms are defined below for the context of this series.

Load Test: A load test is a multi-user test that accurately simulates the expected user community, including user delays. These tests may be executed with differing user loads to find information such as the maximum number of users a site can support whilestill meeting the stated performance goals.

Stress Test: A stress test is any combination of scripts that are played back at a high user load excluding user delays. These types of tests are useful in determining system stability, and functionality under load, but are NOT valid for determining user experience.

Benchmarks: are metrics gathered about system hardware, and supporting software, but NOT application code. For instance, Web server throughput and hits per second when accessing a large graphic can be determined through benchmark testing.

Overview of this Series

The “User Experience, Not Metrics” series will have a new article submitted monthly. All articles will include discussions of the practical applications of the methodology/technique being introduced, real world examples and/or code samples and “Now you try it.” exercises. Each article will be identified as Beginner, Intermediate or Expert level. This level generally refers to the complexity of the code required to accomplish the technique being discussed. The concepts presented in each article will be applicable at all levels of expertise. The first 12 articles have already been identified and are described briefly in the following sections.

Modeling Real Users

One of the keys in determining true user experience is to effectively model actual users and user communities. Most performance tuning approaches today do not account strongly enough for either the distribution of tasks across entire user communities, or the high level of randomness among actual users. The first three articles in the series discuss how to use Rational Test Studio to accurately model both individual users and entire user communities from the application's perspective. Specific topics include:

Part 2: Modeling Individual User Delays
Part 3: Modeling Individual User Patterns
Part 4: Modeling Groups of Users

Meaningful Times

After modeling actual users, it is imperative to capture the actual system/application response time from the perspective of those users. Simply capturing those times is not sufficient. The times are useless unless the patterns of those times can be interpreted. The next three articles discuss how to use Rational Test Studio to capture and interpret true experience times. Specific topics include:

Part 5: What should I time and where do I put my timers?
Part 6: What is an outlier and how do I account for one?
Part 7: Consolidating and interpreting Times

Reports to Stakeholders

As much as I hate to admit it, stakeholders and decision makers need reports on results. I keep trying to convince my clients that all they need from me at the end of a Performance Engineering engagement is a Post-It note with either the words "Go Live", or "Don't" written on it, but they don't seem to think that provides enough value. If you have clients similar to mine, you'll be required to take the vast amount of data collected from a Performance Engineering effort (often several Gigabytes) and consolidate it into concise, yet meaningful report. The articles in this section will discuss what types of tests provide the most value to managers and decision makers as well as how to use the data collected from Rational's Test Manager reports to create multiple run summaries. Specific topics include:

Part 8: What Tests add value to stakeholders?
Part 9: Summarizing across multiple tests with accuracy
Part 10: Creating a Degradation Curve

Advanced Topics

The final group of topics in this series will focus around specific advanced issues that have caused stress to the authors. These articles take the format of case studies. Each case study outlines the specific need of the (unnamed) client in question, the author's thought process to developing a solution, an outline of the potential solutions, and a detailed description of the selected solution. Specific topics include:

Part 11: Handling Secure Session ID's
Part 12: Conditional user path navigation (intelligent surfing)
Part 13: Working with Unrecognized Protocols

Summary

The lesson in this introduction to *The User Experience, Not Metrics* article series is unmistakable; a user's point-of-view is a more reliable measure of website performance than today's customary metrics. This series of articles is designed to teach how multi-user activity can be simulated using Rational's TestStudio and a proven approach to engineering application performance to maximize end-user experience. The articles promise to share valuable information about the how the methodology works and how the Rational toolset is utilized. The articles will even divulge useful tips in getting around those issues that have stumped the experts. I hope your interest has been piqued and that you will return next month for your first dose of *The User Experience, Not Metrics* article series.

Acknowledgments

- The original version of this article was written on commission for IBM Rational and can be found on the IBM DeveloperWorks web site.

About the Author

Scott Barber is the CTO of PerfTestPlus (www.PerfTestPlus.com) and Co-Founder of the Workshop on Performance and Reliability (WOPR – www.performance-workshop.org). Scott's particular specialties are testing and analyzing performance for complex systems, developing customized testing methodologies, testing embedded systems, testing biometric identification and security systems, group facilitation and authoring instructional or educational materials. In recognition of his standing as a thought leading performance tester, Scott was invited to be a monthly columnist for Software Test and Performance Magazine in addition to his regular contributions to this and other top software testing print and on-line publications, is regularly invited to participate in industry advancing professional workshops and to present at a wide variety of software development and testing venues. His presentations are well received by industry and academic conferences, college classes, local user groups and individual corporations. Scott is active in his personal mission of improving the state of performance testing across the industry by collaborating with other industry authors, thought leaders and expert practitioners as well as volunteering his time to establish and grow industry organizations. His tireless dedication to the advancement of software testing in general and specifically performance testing is often referred to as a hobby in addition to a job due to the enjoyment he gains from his efforts.

About PerfTestPlus

PerfTestPlus was founded on the concept of making software testing industry expertise and thought-leadership available to organizations, large and small, who want to push their testing beyond "state-of-the-practice" to "state-of-the-art." Our founders are dedicated to delivering expert level software-testing-related services in a manner that is both ethical and cost-effective. PerfTestPlus enables individual experts to deliver expert-level services to clients who value true expertise. Rather than trying to find individuals to fit some pre-determined expertise or service offering, PerfTestPlus builds its services around the expertise of its employees. What this means to you is that when you hire an analyst, trainer, mentor or consultant through PerfTestPlus, what you get is someone who is passionate about what you have hired them to do, someone who considers that task to be their specialty, someone who is willing to stake their personal reputation on the quality of their work - not just the reputation of a distant and "faceless" company.