



Brochure

# Application performance engineering

Take control with an end-to-end framework for application performance



# Table of contents

**Executive summary**

**Are you seeing two stars instead of five?**

**A framework for better application performance engineering**

**Best practices for application performance engineering**

**A portfolio that is both broad and deep**

**Why HPE?**

**Take control of application performance**



## Executive summary

In today's user-centric world, applications are increasingly at the heart of how your customers experience your products and services. Consistently good application performance is now essential to business success.

This is where the quality of application performance enters the picture. Given the complexities of today's modern application environments, applications should be tested early, often, and thoroughly in the development cycle using processes and solutions to fit your specific needs.

To help your organization meet this objective, this interactive brochure explores an eight-step framework for better application performance. This framework begins with business requirements and culminates in the ongoing optimization of your application performance. With its expansive application software portfolio, HPE covers all of the steps in this framework.

---

## The IDC view

“With such high stakes and high user visibility, poor performance is simply not an option for customer-facing and other business-critical applications.”<sup>1</sup>

**Read the white paper.**

---

<sup>1</sup> IDC white paper, sponsored by HPE, “Driving Business Optimization with End-to-End Performance Testing,” September 2013.

## Are you seeing two stars instead of five?

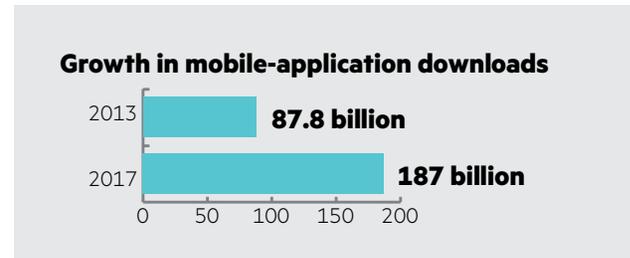
For a mobile or web-based application to be successful, it has to be available at a click of a button—without long wait times, delays, errors, or service interruptions. With today's digitally savvy users, poor application performance is punished immediately with bad user ratings, lost customers, and lost business.

Will your users wait for a slow app to respond? Probably not. Industry research indicates that 25 percent of users abandon a mobile app after just three seconds of delay.<sup>2</sup>

And here's where it gets even harder: The challenge of maintaining peak performance for applications will grow sharply in the years ahead as users make ever-greater use of mobile apps. One sign of what lies ahead: IDC forecasts more than 200 percent growth in mobile application downloads by 2017.<sup>3</sup> Looking out a bit further, by 2020, we will experience 30 times more app refreshes per year compared to 2010.<sup>4</sup>

In this new era, yesterday's piecemeal approaches to application performance management are no longer adequate. To be confident that you can deliver a superior end-user experience, you now need an end-to-end application performance engineering framework that spans the application lifecycle.

As IDC notes: "This concept of a framework for performance testing becomes particularly important as mobile software dominates branding and competitive execution; dynamic, engaging app stores can enable—or decimate—audiences, prospects, and revenue. App store ratings and failures are public and unforgiving."<sup>5</sup>



<sup>2</sup> "First Class Mobile Application Performance Management." The Aberdeen Group. August 2012.

<sup>3</sup> IDC white paper, sponsored by HPE, "Driving Business Optimization with End-to-End Performance Testing." September 2013.

<sup>4</sup> HPE internal analysis, comparing app updates in 2010 to expected app updates in 2020.

<sup>5</sup> IDC white paper, sponsored by HPE, "Driving Business Optimization with End-to-End Performance Testing." September 2013.

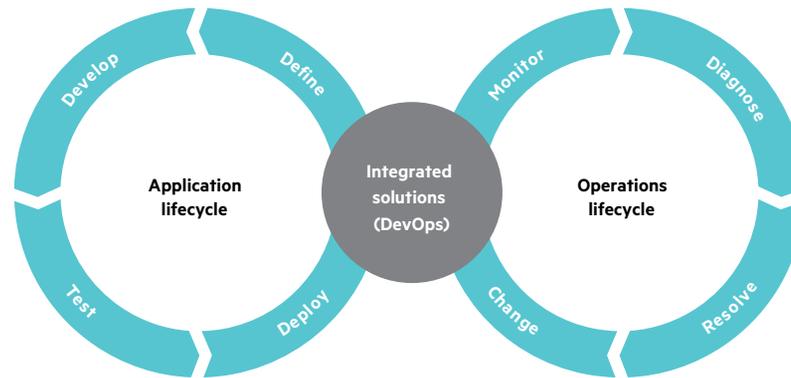
# A framework for better application performance engineering

Better application performance begins with an integrated framework for managing performance across the application lifecycle. This cyclical framework spans from the definition of business requirements to post-production refinements of your application.

Let's walk through the best practices for each of the steps in this framework for better application performance.

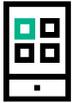
The steps for 5-star apps are:

1. Define
2. Develop
3. Test
4. Deploy
5. Monitor
6. Diagnose
7. Resolve
8. Change



**Figure 1:** Application lifecycle and operations lifecycle must converge.

# Best practices for application performance engineering



## 1. Define business requirements for application performance

From the outset of your project, you want to collect detailed requirements for application performance and clearly define your business priorities, business goals, and any associated service-level agreements (SLAs). These business-driven requirements become high-level measures that govern the pass/fail status of the load testing process.

### Business requirements address such question as:

- How critical is the application for the business and how does performance relate to it?
- How many users must the application support at normal and peak periods?
- What are the user locations, and application timeframe usability?
- What are the minimum and maximum acceptable response times for all business-critical transactions?
- How do the user communities connect to the application?
- What workloads are experienced in production?
- How many transactions per second must the application be able to process?
- What are the special factors about the application that should be considered in terms of security that will affect application performance?

The key is to set clear, measurable goals for performance and load testing that will validate the ability of your application to meet your business requirements. Gathering a complete set of business, technical, system, and team requirements upfront lays the foundation for an effective and successful end-user experience and business growth.



## 2. Develop the application with performance in mind

Once your requirements are in place, the next step is to translate them into development and testing objectives for your application. This step in the process turns business requirements into measurable metrics that cover such factors as response time, load, memory, and CPU utilization. This will provide the developers with guidelines on how to create applications that perform as expected.

Testing early in the software development process is one of the keys to delivering better applications in less time. The goal is to find and fix performance issues at the code and component level. When you catch problems early, you avoid potentially substantial rework later in the development process.

For this reason, it is important to have a performance testing solution integrated with the development tools.



## 3. Test performance continuously throughout the application lifecycle

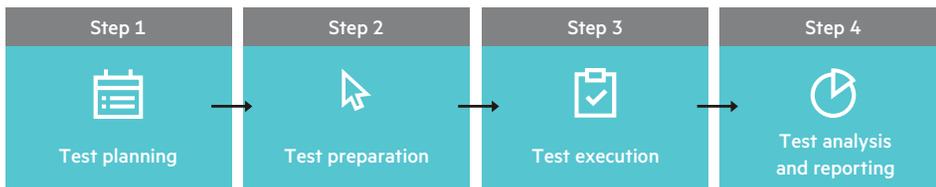
To reduce the risk of costly problems with production applications, you need to test performance throughout the application lifecycle. More specifically, you want to test early and often in the application lifecycle. The goal is to catch any and all performance defects before applications go into production, when defects can be extremely detrimental to your business.

## Brochure

Testing performance manually is not an option. It is important to have a performance testing/engineering process plus a software solution in place.

### Application performance testing process and software testing tools

Application performance tests should leverage robust software load testing tools that closely emulate production environments and the load of tens, hundreds, or even thousands of concurrent users.



**Figure 2:** The testing process

### Here's a look at the high-level steps in the software load testing process:

#### Step 1: Test planning

Successful performance and load testing requires that you develop a thorough test plan. A clearly defined test plan makes sure that the scenarios you develop will help accomplish your load testing objectives. Among other considerations, this planning process identifies the specifics of the test environment, such as platform, servers, network, database, and web services, and your test requirements, such as response time, load, memory, and CPU utilization.

#### Step 2: Test preparation

With software load testing tools, virtual users (Vusers) emulate human users, interacting with your applications. A Vuser script contains the actions that each Vuser performs during scenario execution (transactions). Once the script is created, you can execute the script with few virtual users to test the script and adjust any discrepancies before executing the full load testing. Before executing the test, you must also determine how the test will be executed: how many transactions will be tested simultaneously by how many users in a period of time and its intervals. That is why it is important to collect all this information in the requirements phase.

#### Step 3: Test execution

You can emulate user load on the application by instructing multiple Vusers to perform tasks simultaneously. You can set the level of load by increasing and decreasing the number of Vusers that perform tasks at the same time. As you execute the test you can monitor the infrastructure to understand how the load will impact the production components.

#### Step 4: Test analysis and reporting

The testing solution should incorporate analysis capabilities that help you quickly determine which transactions passed or failed your set service-level objectives and help you identify potential causes of failure. A strong analysis engine should help you slice and dice data in many ways to easily pinpoint the root cause of problems.

#### Points to consider for performance testing:

**Mobile characteristics**—When you're testing mobile apps, you need to consider the differences between mobile apps and web apps. For example, compared to web apps, native mobile apps have greater access to all of the functionality on the mobile device, and the testing process needs to take this into account.

**Service virtualization**—New applications are increasingly built on distributed service architectures, including mobile and cloud-based services that are not in the test lab. In addition, quality assurance (QA) specialists often have limited access to production systems for app testing. Service virtualization allows you to bypass these roadblocks.

Service virtualization enables your teams to create, develop, and test software against virtual services that simulate real service behavior with no constraints, available anytime. This capability helps you keep your project on schedule even when you can't develop or test the real versions of applications, dependent systems, and services.



#### **4. Deploy fully tested applications**

When you incorporate robust load testing throughout your development process, you can deploy your applications with confidence. You know that your applications can stand up to your expected user loads, network conditions, and service variables. Leveraging automated solutions to move code to production will reduce the risk of production issues. Plan to leverage the previous defined scripts used for test to monitor the application in production.



#### **5. Monitor applications and the end-user experience**

When applications go into production, the path to high performance on an ongoing basis lies in the aggressive monitoring of the applications and the end-user experience, and then feeding performance data back to your development teams. This proactive work is one of the keys to avoiding emerging performance issues with production applications.

Throughout the monitoring process, the goal is to fully understand the end-user experience and rapidly diagnose any performance issues. To meet this goal, your monitoring toolset should include capabilities for monitoring the real-user experience, so you know what's actually happening in the world of end users.

Application performance issues should be captured in an incident management solution.

Look for solutions that can provide the real user experience data, so it can be leveraged as baseline for new development, changes, and test.



### 6. Diagnose the root cause of issues

In addition to monitoring applications and end-user experience, your performance management toolset should be designed to accelerate root-cause analysis. This work leverages information gained from event monitoring, performance monitoring, load testing, and network monitoring. In these efforts, your monitoring teams can leverage assets from the testing process, such as application diagnostic tools.



### 7. Resolve application performance issues

When you detect application performance issues and identify the root causes, the information you gain is fed back to your development team to fix issues with the underlying code. In some cases, application performance problems might trigger the redesign of the application.



### 8. Change the application based on performance data

The application lifecycle doesn't end with the deployment of a production application. Production data should be continually fed back to your development and testing teams so they can validate the performance of the application, retest changed and modified software, and look for additional opportunities to optimize the application code.

# A portfolio that is both broad and deep

With a portfolio that is both broad and deep, HPE is uniquely positioned to deliver your end-to-end solution for application performance engineering and management.

The HPE performance engineering portfolio supports:

- Multiple technologies
- Integration with various HPE solutions and tools from other vendors
- Integrated developer tools
- Root-cause analysis

- DevOps capabilities
- Analytics
- Metrics/executive scorecards
- Analysis on network behavior and characteristics
- Service virtualization

Key products in the HPE performance engineering suite include HPE StormRunner Load, HPE LoadRunner, HPE Performance Center, and HPE Performance Center on SaaS.

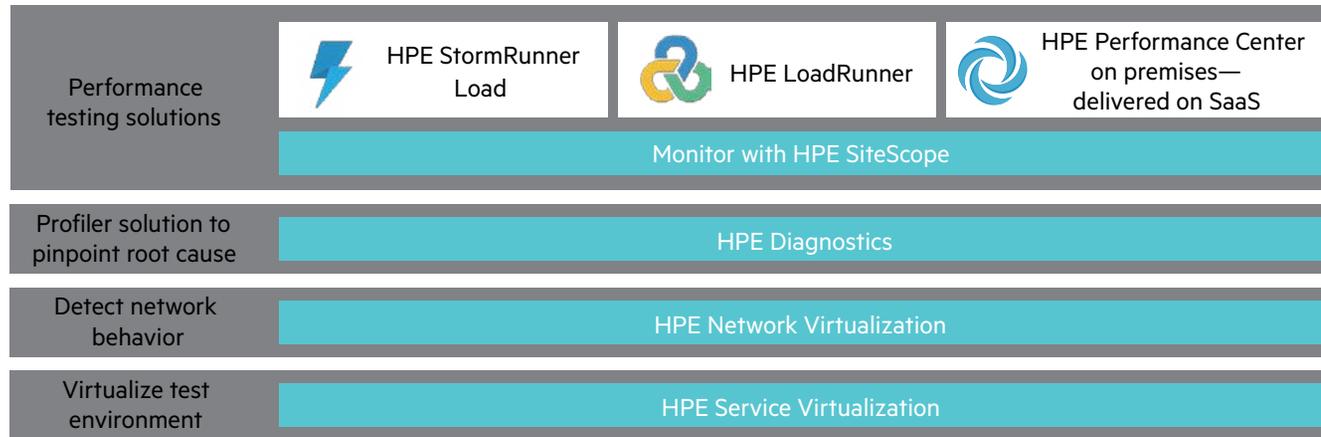


Figure 3: The HPE performance engineering portfolio



### **HPE StormRunner Load: Agile Cloud testing**

#### **Simple, smart, and scalable**

With HPE StormRunner Load, your software delivery team now has fast, easy access to cloud-based load testing resources that are always just a click away. It's a simple, smart, and scalable approach to testing web and mobile apps. Try it now: [saas.hpe.com/software/stormrunner-load](https://saas.hpe.com/software/stormrunner-load).

### **HPE LoadRunner: On-premises project load testing**

#### **Easy, fast, powerful—and now FREE**

HPE LoadRunner is the industry standard for performance testing. This application load testing software tool helps you gain an accurate picture of end-to-end system performance and identify and resolve issues before your application goes live. With HPE LoadRunner, you can dramatically reduce the amount of time and skill required for scripting. HPE LoadRunner is now offered free for 50 virtual users for an unlimited time. Try it now: [saas.hpe.com/software/loadrunner/try-now](https://saas.hpe.com/software/loadrunner/try-now).

### **HPE Performance Center: enterprise performance engineering solution**

#### **Enable enterprise-wide performance validation**

HPE Performance Center software is an enterprise-class performance engineering software suite, designed to facilitate standardization, centralization, global collaboration, and management of a performance engineering center of excellence. Try it now: [saas.hpe.com/software/performance-center](https://saas.hpe.com/software/performance-center).

Built on HPE LoadRunner, HPE Performance Center helps you plan and execute tests across multiple global projects. HPE Performance Center is now offered free for 50 virtual users for an unlimited time.

### **HPE Performance Center on SaaS**

#### **Leverage software as a service (SaaS)**

HPE Performance Center on SaaS provides the capabilities of HPE Performance Center as an on-demand, SaaS solution for application performance testing. It eliminates the need to have hardware on your premises. [Click here to try it now.](#)

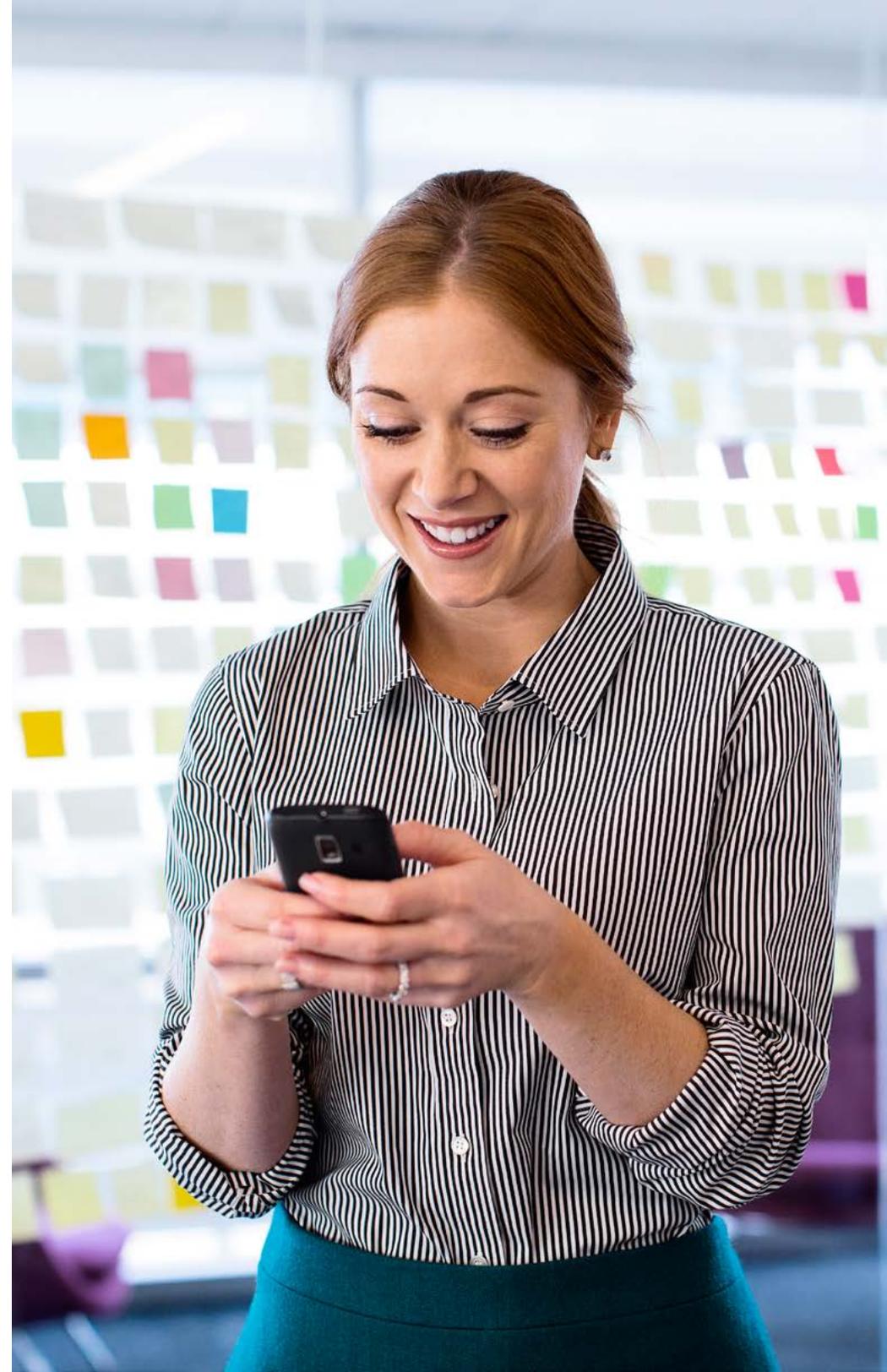
## Brochure

### **HPE Network Virtualization: Detect network behavior Predict network conditions for a complete user experience**

HPE Network Virtualization is designed to enable precise network performance testing scenarios that deliver highly accurate and reliable performance results. Effective insight and analysis of those results are critical to affect meaningful and cost-effective application or infrastructure change. Learn more at: [saas.hpe.com/software/network-virtualization](https://saas.hpe.com/software/network-virtualization).

### **HPE Service Virtualization: Simulate components or services Accelerate development and test with an end-to-end application environment**

Simulate a service's behavior in a production environment. This simulation software enables in-house development and testing teams to keep to their schedules regardless of access to production systems. Learn more at: [saas.hpe.com/software/service-virtualization](https://saas.hpe.com/software/service-virtualization).



# Why HPE?

## An end-to-end solution

HPE is uniquely positioned to deliver your end-to-end solution for application performance management. The HPE solution spans the application lifecycle—from application design to development, testing, monitoring, and bringing data from operations back to requirements.

## Integration

For end-to-end application performance management, you need more than a collection of point tools. You need a comprehensive, tightly integrated solution that spans the application lifecycle. That's the HPE portfolio.

## Technology

HPE application performance management solutions support a wide range of protocols, monitoring tools, automatic correlation, and async capabilities, a rich GUI for script creation and editing, and VuGen extensibility.

## Leadership

HPE has more than 20 years of experience in application performance testing tools, more than 5,000 performance engineering customers, and a market leadership position.

“An ‘end to end’ application performance testing strategy becomes particularly important given complex sourcing, including offshoring and outsourcing, open source usage, regulatory compliance, and emerging new development paradigms.”

– IDC<sup>6</sup>



Figure 4: Why HPE performance engineering

<sup>6</sup> IDC white paper, sponsored by HPE, “Driving Business Optimization with End-to-End Performance Testing,” September 2013.

# Take control of application performance

For more information, visit the:

- [HPE blog for performance testing](#)
- [HPE performance testing forums](#)
- [HPE LoadRunner Twitter](#)
- [HPE LoadRunner Facebook group](#)

Learn more at

[saas.hpe.com/en-us/software/performance-engineering-virtualization](https://saas.hpe.com/en-us/software/performance-engineering-virtualization)



Sign up for updates