

Beyond Black Friday

Be ready for anything,
anytime, anywhere



Contents

- 3 Any day could be Black Friday
- 4 Set your site up for unexpected success
- 5 Poor performance could cost you big
- 6 Meet testing needs with OpenText™ Core Performance Engineering
- 7 Maximize possibilities, minimize challenges
- 8 Hit the peaks: Better performance at crucial times
- 9 Five tips for better testing with OpenText™ Core Performance Engineering
- 10 Doing things manually sucks

Any day could be Black Friday

Remember when the Walmart website crashed during Black Friday? Many customers were unable to complete their online purchases and missed out on deals.¹ When major retailer sites crash at key moments, revenues are lost, brands are damaged, and customers are annoyed and disappointed.

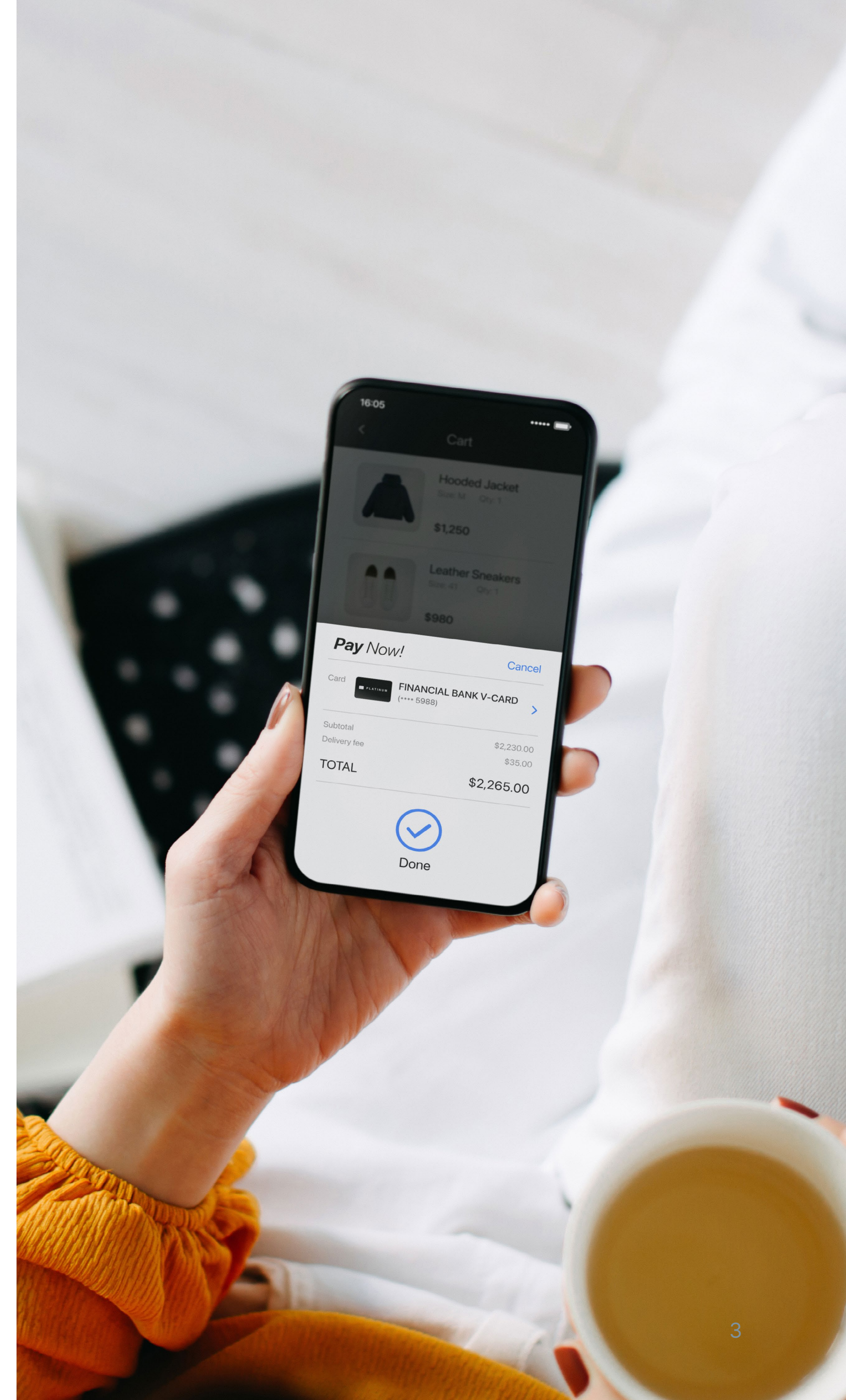
As internet sales events become more common, so do potential profits. For [Black Friday](#) 2023, total online sales in the United States amounted to \$9.8 billion.² Online shopping is forecasted to grow by nearly 11 percent between 2023 and 2027.³

Spikes in demand don't always follow a calendar. A run on your website could come any day, at any time, for really any reason. Maybe a celebrity was seen with your product, a major publication put out an outstanding, unexpected review, or it just became the current hit meme. Failure to prepare means missed opportunities and damaged reputations.

¹ The Sun, Down and Out: Walmart site & app crash just before revealing exclusive Black Friday deals – shoppers crying 'already missing out', 2023

² FXCintelligence, 2023 Black Friday sales break records despite cost-of-living rise, 2023

³ Statista, Online shopping behavior in the United States – statistics & facts, 2023



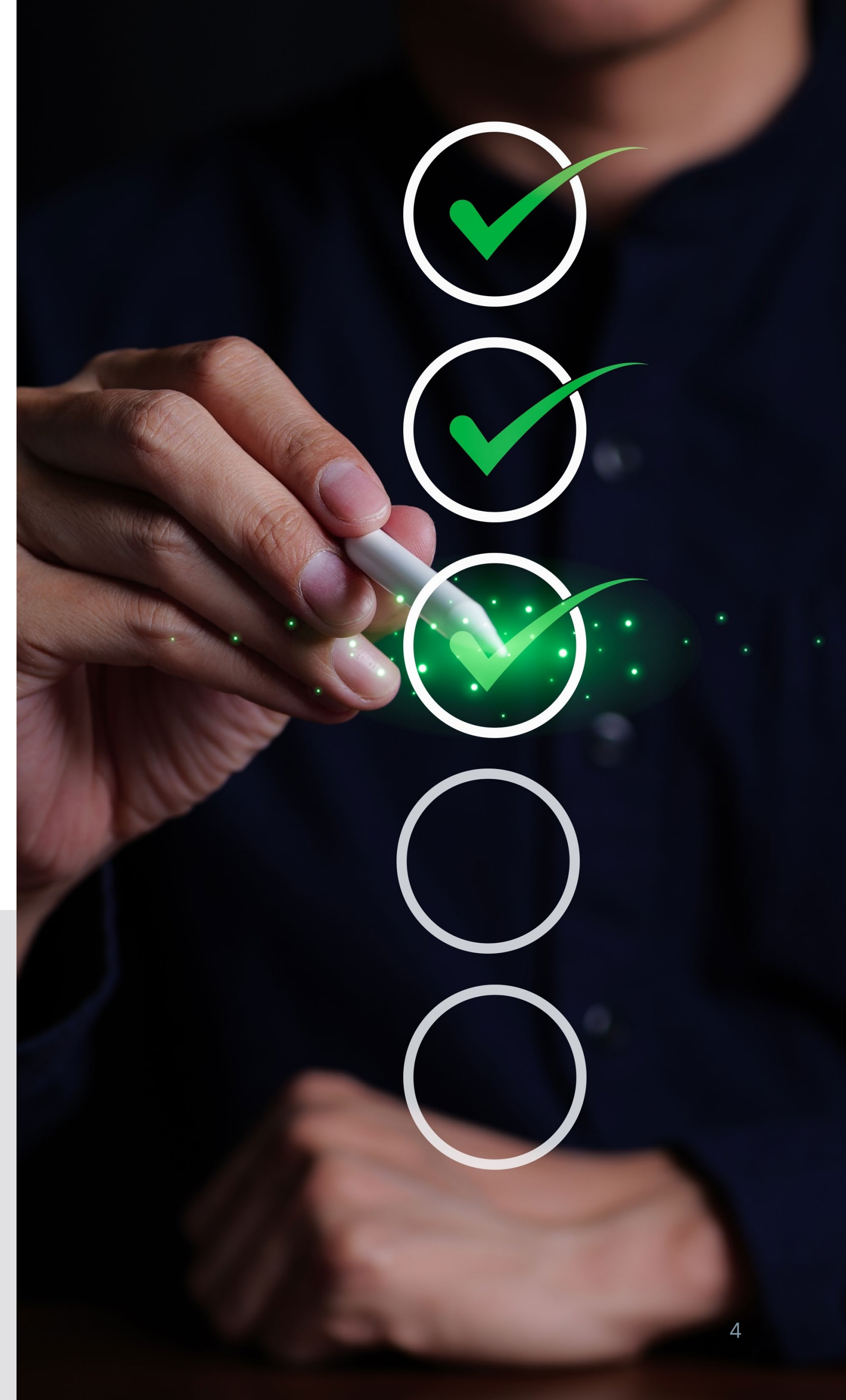
Set your site up for unexpected success

Online retailers must plan for the big events they know about and prepare for those they don't. A robust performance testing strategy combined with the best tool out there can help.

We can point the way to performance testing salvation and ensure your online retail site is all boom, no bust.

Let's go!

**If you can't predict demand every time,
(spoiler: you can't) you need
to be ready all the time.**



Poor performance could cost you big

\$9K

in lost revenue per minute due to IT downtime.⁴

\$100K

average hourly cost of infrastructure failure.⁵

83%

of users expect a site to load in 3 seconds or less.⁶

90%

of users don't return to a website after a bad experience.⁷

48%

of users think that companies without a mobile application don't care about them.⁸

40%

of adults use mobile applications to search for B2B products.⁹

85%

of users think a company's mobile app should be better than their desktop site.¹⁰

2.6 seconds

for users to make the first conscious impression of a website.¹¹



4 Solarwinds Pingdom, Average Cost of Downtime per Industry, 2023

5 Medium, Here's how much downtime is really costing your business, 2023

6 WebFX, 14 Site Speed Statistics To Boost Your Loading Times, 2024

7 Website Builder, 30 Astonishing Website Design Industry Statistics for 2024

8 Ibid.

9 Ibid.

10 UXCam, 20+ Powerful UX Statistics To Impress Stakeholders 2024, 2024

11 Website Builder, 30 Astonishing Website Design Industry Statistics for 2024

Meet testing needs with OpenText™ Core Performance Engineering

You need to:

- Simulate realistic global user scenarios.
- See how mobile users impact your backend.
- Emulate the impact of various network types.
- Ensure your site performs when user traffic peaks.

With a pay as you need model, [OpenText Core Performance Engineering](#):

- Simplifies how you plan, run, and scale your testing for web and mobile apps.
- Simulates any network type with built-in network emulation.
- Scales to more than five million virtual users.
- Uses rich analytics to create breakdown reports.



Maximize possibilities, minimize challenges


OpenText Core Performance Engineering makes peak testing easier, faster, and more cost-efficient. With cloud capabilities, it's available on demand and is highly scalable.

Why do I want OpenText Core Performance Engineering?

With it, you can:

- Test quicker and identify problems faster through root cause analysis.
- Save testing money and time without additional infrastructure.
- Reduce hardware maintenance with a flexible testing model.
- Optimize application performance before deployment.

If you don't test, your customers are discovering your errors, aren't they?



“I want peak testing to be complicated, slow, super-expensive, and hard to get.”

- Said no one, ever.

Hit the peaks: Better performance at crucial times

So how do you use OpenText Core Performance Engineering to ensure site stability?

Don't leave testing late.

Performance tests are an integral part of the process, not a just-before production or we-already-have problems thing. Allow time to fix what you find.

Decide what you need.

Temporarily increasing infrastructure to match large user loads is easier in the cloud, especially for subject matter experts. Be sure your scaling configuration factors in both performance and costs.

Test on what you know.

Use the infrastructure you prefer, or what you have in production, at the best moment to run the tests. That may be during late hours using engineers in a different time zone. A five-hour difference is usually plenty.

Prioritize to maximize.

No time to test everything? Choose carefully. What will be the most visited part of your site? Links to articles? The product checkout?

Monitor performance.

Stay updated on the health of your infrastructure. If there's a problem, be the one to find it first.

Be realistic.

Run real-world loads for your business. Testing too small means going dark on Black Friday, while being too ambitious means unnecessary stress.

Five tips for better testing with OpenText Core Performance Engineering

1

Be browser-agnostic

Customers use [multiple browsers](#), so your site must deliver a consistent user experience on all of them. However, manually testing devices takes time and costs big.

2

Think always online, everywhere

Customers like [mobile-friendly](#) sites that load quickly and perform well on any device across the globe. If they take more than three seconds? Bye, bye.

3

Beat the peaks

Seems obvious, but showtime is no time to rehearse. Likewise, don't wait for seasonal retail spikes to see if your websites can cope.

4

Use that self-driving test lab

Because the service is hosted in the cloud, teams can automatically create test infrastructure without managing or maintaining controllers or load generators.

5

Go large—scale up, and up, and up

When your big moment arrives, be ready. Demand can come from anywhere and reach new heights. Make sure your sites and applications can handle the pressure of big numbers.

A hand holding a white smartphone against a teal background with a large white cloud.

Doing things manually sucks

Use OpenText Core Performance Engineering to automatically:

- **Create extra infrastructure**, increase testing speed and effectiveness, and reduce development and maintenance costs.
- **Create hundreds of load generators** on demand in more than 30 global cloud regions. Create more than five million virtual users—and fast.
- **Plan, run, and scale** your web and mobile application testing to uncover the root cause of performance issues.
- **Emulate multiple user populations** and re-create real-world end-user conditions.

Get all of the above—without expensive test infrastructures.

Think boom—not crash!

Learn more about our [extremely scalable, cloud-based](#) load and performance testing solution, OpenText Core Performance Engineering.

[OpenText Core Performance Engineering data sheet ›](#)

[OpenText Core Performance Engineering brochure ›](#)

[OpenText Core Performance Engineering web page ›](#)

About OpenText

OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit opentext.com.

opentext.com | [X \(formerly Twitter\)](#) | [LinkedIn](#) | [CEO Blog](#)