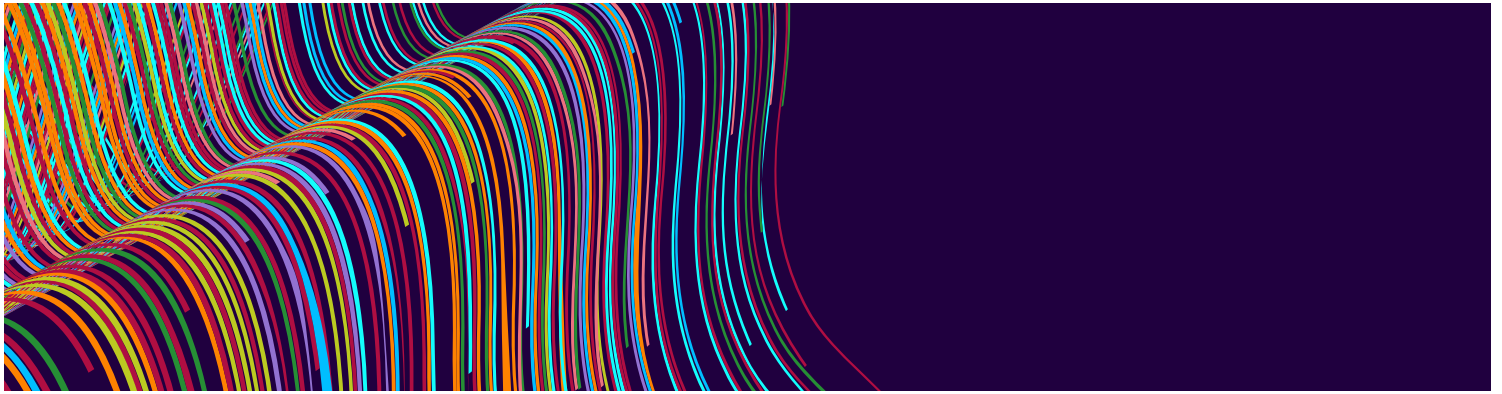




The Three Dimensions of Monitoring Digital Customer Experience

Understanding the elements that go into the monitoring of the digital customer experiences you deliver.

As companies move more and more of their customer interactions to the web and mobile devices, the digital customer experience increasingly becomes the primary way they interact with their customers and prospects. But to truly grasp the importance of delivering a great digital customer experience, you have to understand exactly what the term means, what dimensions compose a digital customer experience, and what Key Performance Indicators (KPIs) to track to determine the Quality of Service you are providing.

What does digital customer experience (DCX) even mean?

Every interaction a customer has with your website, mobile app, or other digital touchpoint leaves behind an impression. When you combine these different impressions, the collective mosaic they create composes your overall digital customer experience.

Why does monitoring my DCX matter?

Ready or not, digital is now the primary relationship many customers have with your business. Your websites and mobile apps are now your marketing/service channels, brand experiences, and often important sources of revenue. DCX issues can have real business impacts on

revenue, customer satisfaction, and brand reputation. Negative digital experiences, such as if your website goes down or your mobile app crashes, don't just affect the engineering teams that run your digital channels, but can go a long way toward determining the success of the overall business.

Quality of Service dimensions for a great digital customer experience

Because the quality of your digital customer experiences can have such a profound impact on your business, it's critical to properly measure the service-level quality of the digital experiences you deliver across three main dimensions:

1. **Availability**—Is it up and running?
2. **Functionality**—Is it working right?
3. **Speed**—Is it working fast enough?

This order helps to triage any problems you encounter. If your page isn't up, then functionality and speed don't matter. If your page throws errors and is non-functional, then speed isn't so important. But even when the site is up and working properly, chronic slowness corrodes the user experience and causes user frustration.



Applying the three dimensions of DCX

For deeper insight, let's apply these dimensions to some sample contexts to examine the service-level quality of different digital experiences and identify the KPIs most relevant to your business and the digital customer experience that you're delivering.

VIDEO STREAMING IN A DESKTOP WEBSITE:

1. **Availability**—Does the website load? Is it crashing? Are the backend video delivery services up and running?
2. **Functionality**—Did the website crash when the user tried to log in to view the video? Did the player load? Is the video loading? Are ads loading?
3. **Speed**—Did the video stutter? Did the bitrate drop? How long did it take for ads to load?

A NATIVE MOBILE FOOD-ORDERING APP FOR IN-STORE PICKUP:

1. **Availability**—Is the app being launched? How often does the app crash? Is the store printer working?
2. **Functionality**—Was the user able to purchase? Did the order get created correctly? Did it route through the printer correctly?
3. **Speed**—Did the mobile app work smoothly on order confirmation? Was the food ready when the customer came to pick it up?

AN ENTERPRISE TECHNOLOGY COMPANY OFFERING A RICH WEB APPLICATION AS A SERVICE:

1. **Availability**—Is the web application up? Does it meet the contractual uptime SLA?
2. **Functionality**—Are users able to log in? Can they get to the main dashboard? Can they complete their critical activities?
3. **Speed**—Does the web app work smoothly? Does it get slower for VIP customers with lots of data in the service? Are users bouncing prematurely out of frustration?

Unsurprisingly, different apps for different purposes have different critical factors. Starting with these user- and business-oriented metrics can help you understand the digital experience delivered by your technology teams.

Establishing underlying Quality of Service metrics

While these dimensions help answer the high-level questions important to the business, we also want to drill down to the underlying technology metrics that feed into these dimensions. These lower-level metrics granularly address concrete technology layers that teams can work on. Examples include:

Availability

- User throughput
- Global service uptime
- Frontend uptime
- Backend uptime
- Mobile app launches
- Mobile app crash rates
- API uptime
- Third-party service uptime
- Server uptime

Functionality

- Frontend error rates
- Backend error rates
- API error rates
- Third-party error rates
- Key user-transaction errors
- Container-management errors
- Infrastructure errors

Speed

- Frontend response time
- Backend response time
- API response time
- Application transaction times
- JavaScript execution time
- Database query times
- Resource utilization
- Container health
- RAM, CPU, network latency
- Third-party service response times



The wide breadth of these health metrics reflects the increased complexity of modern application architectures and their underlying technology stack. Degradation of your digital customer experience can result from many different root causes, emanating from different layers of your technology stack, from the frontend to the backend/database to the supporting infrastructure.

- **Frontend example:** Unnecessary AJAX calls in a single-page application. A particular user interaction in a single-page application triggers multiple, duplicated AJAX requests to a backend service, prolonging load times for users of a desktop website.
- **Backend/database example:** A bad database query does a full select statement. This slows down the backend API service, which makes it unresponsive to the native mobile app requesting the data. The request times out, resulting in an intermittent mobile app crash.
- **Infrastructure example:** Incorrect container service configuration leads to under-provisioning of infrastructure services. Because there are not enough resources, requests cannot be handled promptly, overloading the load balancer and bringing down the system—the whole stack goes down.

Next steps

Technology teams can no longer limit themselves to just building and managing their part of the stack; they now bear responsibility for how their work impacts customers and the business. Monitoring your digital customer experience helps development and operations teams find and solve problems anywhere they're located.

Now that you understand the dimensions and KPIs that you want to track, it's time to get started on actually doing something about them. Check out our technical guide for Best Practices for Monitoring Digital Customer Experience for help optimizing your full technology stack and integrating DCX monitoring into your workflows. Then you're ready to show the impact of these improvements to the larger business.

Managing the sea of data

With KPIs through the entire technology stack, the challenge arises around identifying the most important ones. Similar to the hierarchy of availability, functionality, and performance, we can identify the most meaningful underlying metrics to separate the signal from the noise.

The metrics closest to the user and the business best represent what's actually happening. It's a more authoritative test of the end result delivered by your technology stack. If something is broken for a customer, that supersedes any positive indicators. This often means starting with actual **page load times**, **mobile app launches**, or **scripted user-action monitors** to understand how an app is working in the wild.

Conversely, metrics deeper in the technology stack can serve as early warning indicators. They're noisier and less explicit, but they can hint at potential issues before they bubble up to your customers. In addition, these deeper metrics can help troubleshoot the root cause of customer-impacting issues. Use them to find those **bottlenecking microservices** or **misconfigured infrastructure** resources before those problems make their way to your customers.