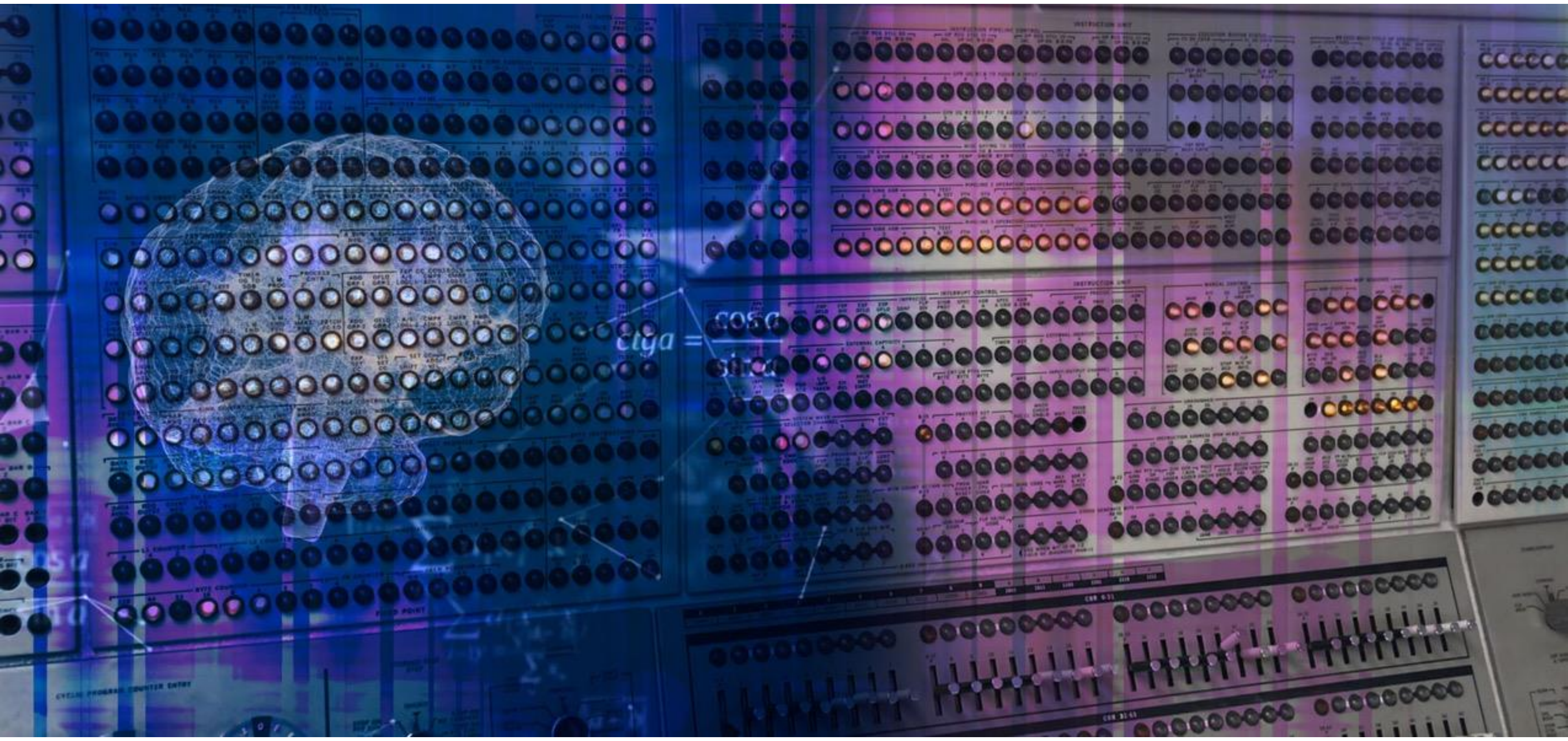

White Paper

How to Optimize Digital Experience with Service-Level Objectives

Bringing System Reliability Engineering to Digital Experience Management

Jason Bloomberg
President, Intellyx

April 2021

Optimizing how people perceive their interactions with a web page, application, or other software interface – what we call Digital Experience (DX) – is a fundamental goal of any software effort.

Digital Experience Management software gives organizations the ability to monitor and manage DX, but optimizing DX means taking into account all the metrics that measure the experience, along with the cost and development time commitments necessary to achieve DX goals.

System Reliability Engineering (SRE) gives us a path for achieving such optimizations. By establishing and measuring Service-Level Agreements (SLOs) for the metrics that impact the DX, organizations can leverage Digital Experience Management to optimize their DX and thus achieve the business goals of their software efforts.

The Importance of Digital Experience

Digital Experience (DX) represents the totality of all human reactions to interactions with an organization via any or all digital touchpoints. DX comprises the more established notions of *Customer Experience* (CX) and *Employee Experience* (EX) – frequently lumped together under the *User Experience* (UX) umbrella.

DX, however, is more than the subset of UX that pertains to digital touchpoints. For today's digitally transformed organizations, *all* touchpoints are either directly or indirectly digital. DX, therefore, takes on a more strategic meaning, representing the totality of experiences that drive customer satisfaction, impact brand reputation, and by extension, the overall profitability of an organization.

> Digital Experience is more than the subset of User Experience that pertains to digital touchpoints. For today's digitally transformed organizations, *all* touchpoints are either directly or indirectly digital.

The stakes couldn't be higher, as DX ties directly to business key performance indicators (KPIs). For example, the BBC found that "for every additional second a page takes to load, 10 per cent of users leave."

Similarly, Google found that when a site meets key DX thresholds, "users are 24% less likely to abandon page loads." Similarly, its data showed "22% less abandonment for news sites and 24% less abandonment for shopping sites" when those thresholds are met.

In fact, there are many different technical characteristics of user interactions with digital touchpoints that roll up into the overall DX. Today's web sites and mobile apps contain numerous elements that may come from different servers, APIs, and third parties – and all of them must perform to a given standard for the DX to be sufficient to achieve the goals of the organization.

The Strategic Role of Digital Experience Management

To meet this need, *Digital Experience Management* (DEM) technologies have exploded on the marketplace, combining the capabilities of several point products:

- ***Real User Monitoring (RUM)***: Collecting and analyzing information about how real users experience a digital touchpoint, including web sites, mobile apps, or other touchpoints.
- ***Synthetic Monitoring***: Collecting and analyzing information on the performance of digital touchpoints in response to artificially generated 'synthetic' digital traffic.
- ***Performance Optimization***: A range of techniques and best practices for analyzing and improving the speed and performance of digital touchpoints.

By combining these three sets of capabilities, DEM provides a customer (or employee) centric approach to observability that helps organizations optimize user engagement by identifying any issues that might impact the DX.

Digital Experience Management is more than simply a combination of technical metrics and web performance recommendations. It empowers organizations to measure and optimize the end user's experience with all digital touchpoints, and thus with the organization's brand.

DEM, however, is more than simply a combination of technical metrics and web performance recommendations. It empowers organizations to measure and optimize the end user's experience with all digital touchpoints, and thus with the organization's brand.

Optimizing DX requires a delicate, often complex balancing act that combines baselining and measurement of website components that change as frequently as a business updates its inventory.

Bringing DEM to System Reliability Engineering

Succeeding with this balancing act requires a combination of modern DEM with the latest best practices from System Reliability Engineering (SRE), extended to the realm of DX. (SRE can also stand for Site Reliability Engineering, or System/Site Reliability Engineer.)

SRE is the discipline of applying software engineering principles to infrastructure and operations problems, in particular, creating and maintaining scalable and reliable software systems.

SRE brings a level of formality to long-standing practices of operations by establishing quantifiable measures of reliability in order to objectively manage operations to achieve the KPIs for reliability and associated metrics.

Reliability, however, isn't the only priority. Given today's organizations are aligning all of their IT operations to meet the needs of customers – in other words, becoming digital organizations – the practice of SRE must necessarily treat DX as high priority as well.

Any distinction between DEM tools as being 'front end' while SRE tools are 'back end,' therefore, are obsolete. DX depends upon the end-to-end, back-to-front performance and reliability of the entire IT environment. The practice of SRE must reflect this fact.

Service-Level Indicators for DEM

SRE begins with a formal statement of users' expectations about any particular dimension of the behavior of a site, app, or other piece of software: what we call the *service-level indicator*, or SLI. The SLI is the proportion of valid events that were 'good,' expressed as a percentage.

For the SRE focused on IT infrastructure, 'good' can refer to availability, latency, the freshness of the information provided to users at the user interface, or other key performance metrics that are important to the business. For example, an SLI might state that 99.9% of valid requests for the page index.html were successful (returned a 200 'OK' HTTP code).

Today's SREs, however, must focus more broadly on DX. The RUM and synthetic monitoring of sites and mobile apps that DEM brings to the SRE's toolbox are critical for measuring additional indicators that impact the DX.

We call these additional criteria *web vitals*. Web vitals and other user-centric performance metrics are an advancement from traditional Page Load Time (PLT), which was effective in the early days of the web, when pages were mostly HTML and images.

Today, web vitals include:

- *Largest Contentful Paint (LCP)* – How long it takes for the navigation from one page to another to appear complete. This value is typically less than the PLT because there may be other page elements off the screen that have yet to load.
- *Cumulative Layout Shift (CLS)* – How frequently a page layout shifts unexpectedly while loading. The more dynamic elements appear on a page, the more likely those elements will move around while the page is loading.
- *First Input Delay (FID)* – The time interval between a user's first interaction with a page or other interface and when the browser (or app) is able to begin its response to the interaction. In other words, the time interval between clicking something and the page or app reacting to the click.

Each SLI provides a guideline for each dimension of DX an organization wants to measure for a given user journey. The SLI for these metrics might state that 99% of valid requests to a particular page achieved the targets for the web vitals in question.

The Service-Level Objective: The Practitioner's View

Once the ops team has specified the SLIs important to the business, they must make the appropriate decisions about measurement and validity. These thresholds, in turn, inform the *service-level objectives* (SLOs) for the site or app in question.

The SLO for a system is a precise numerical target for a key metric. For SREs who are focusing on infrastructure, the availability of the system is perhaps the most important metric to have an SLO for. When SREs focus on DX, the SLO will represent a combination of web vitals.

To define your SLO, start with your SLIs. Make sure they each have a success criterion. The SLO, in turn, specifies the proportion of SLI events that were good.

For example, a typical availability-focused SLO might state that 99.9% of the valid requests for the page index.html over the last 30 days returned a 200 'success' code in 150 milliseconds or less.

By leveraging DEM, we can expand the focus of the SRE beyond availability to whatever combination of web vitals are important to achieve the customer experience goals of the organization. The resulting SLOs typically include all relevant web vitals, as a poor result for any single web vital will adversely impact the DX overall.

By leveraging DEM, we can expand the focus of the SRE beyond availability to whatever combination of web vitals are important to achieve the customer experience goals of the organization. The resulting SLOs typically include all relevant web vitals, as a poor result for any single web vital will adversely impact the DX overall.

For example, an SLO might state that 99% of the valid requests for the page index.html over the last 30 days achieved an LCP of 2.5 seconds or less, a CLS score of .15 or better, and an FID of less than 300 milliseconds.

Given the capabilities of a DEM like Splunk's, all of the components of such an SLO are continually measurable, and thus it is possible to calculate automatically whether or not the page in question is meeting its SLO on an ongoing basis.

Don’t Confuse Optimal with Perfect: Introducing the Error Budget

The fact that there are so many variables that make up DX today underscores an important consideration: the *optimal* DX will never be *perfect* DX.

It’s important to note that the SLO will never specify perfect behavior – nor should it. In the example above, the goal is for 99% of the page requests in question to meet their respective targets. The remaining 1% we call the *error budget*.

When we say ‘error budget,’ we’re not referring to an error in the sense of a mistake. Rather, our SLOs always allow for a certain number of requests that fail to meet the given targets, regardless of the underlying causes of such failures.

The point of error budgets, and indeed the core principle of SRE generally, is that *error budgets should never be zero, because there are always tradeoffs between SLOs and the cost and development time needed to achieve them*.

> Error budgets should never be zero, because there are always tradeoffs between SLOs and the cost and development time needed to achieve them.

If we had an SLI that called for an LCP of 2.5 seconds or better, for example, we might consider trying to lower this target to, say, 1.5 seconds. In order to achieve that goal, we might need to spend $10 million and an additional 1,000 person-hours rewriting our page delivery code to more efficiently serve pages with numerous third-party elements. Is it worth the trouble?

The answer: there will always be such tradeoffs, and your SLOs must reflect those tradeoffs. Put another way, you always have an error budget to spend.

Managing Error Budgets to Optimize Performance with Splunk DEM

Once you've defined your SLOs, you should frame all discussions about whether a particular system is maintaining an adequate DX in terms of whether it is meeting its SLOs. By measuring web vitals, ops teams can understand how end-users perceive their services and prioritize engineering efforts accordingly.

In particular, it's always important to frame this discussion in terms of core business priorities like cost, the criticality of individual pages, or the performance of services like login and authentication.

Based upon this calculation, define the lowest level for each web vital that the business is willing to pay for in order to set its desired SLO for the individual pages or services, or for the site or app as a whole.

Based upon this SLO, then, the ops team and its stakeholders can make fact-based judgments about which pages or services negatively impact end-user experience the most, identify which page resources cause the poor performance, and then prioritize engineering efforts to fix the problem.

The ops team can make fact-based judgments about which pages or services negatively impact end-user experience the most, identify which page resources cause the poor performance, and then prioritize engineering efforts to fix the problem.

Both RUM and synthetic monitoring are critical for making these quantitative judgments about web vitals. Improving page performance overall requires both real user field data as well as synthetic lab data.

Splunk RUM and Synthetic Monitoring collect lab and field data that connects into backend services. While web vitals are clearly important to DX, Splunk DEM extends past front-end concerns to back-end service visibility and infrastructure health overall, helping engineering teams correlate and quantify the performance impact of backend services to the end-user experience.

Splunk's intelligent optimization engine offers over 50 modern performance metrics, filmstrips, and screen recordings to show the precise user experience on web or mobile. It also automatically groups performance defects by level of severity and offers guided advice on how to remediate performance problems to help developers improve the user experience while simultaneously improving DX.

Because Splunk DEM extends visibility from client-side performance to backend services via distributed tracing, it helps engineers understand how problems with backend services like slow database queries impact DX, reliability, availability – and in turn, overall system performance.

The Intellyx Take

The first reaction many people have when they realize they can monitor web vitals in real-time is to set unrealistic goals for those metrics. After all, if a two-second PLT or LCP is good, then wouldn't one second be better?

The reality is more complex. Not only are there multiple metrics, where any one of which dropping too far below its target will impact the overall DX, but we also have cost, development time, and service criticality considerations as well.

Such tradeoffs are the reality of modern computing. Software running in production is never perfect, and unattainable targets do us no good. What we need is a measurable approach for optimizing the overall result. This is why DEM is so important.

The most important overall result for any software effort is, in fact, the DX. How customers, employees, or other end-users actually experience their interactions with the software are the standard we must measure all our efforts by.

The good news: in many cases our end-users won't even notice the imperfections in our delivery of software value. Error budgets aren't about presenting shortcomings that adversely impact the DX.

On the contrary, the ideal error budget describes just how many compromises we can make *without* impacting the DX. That's a goal we can all agree on.

About the Author: Jason Bloomberg

Jason Bloomberg is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

He is founder and president of Digital Transformation analyst firm Intellyx. He is ranked #5 on Thinkers360's Top 50 Global Thought Leaders and Influencers on Cloud Computing for 2020, among the top low-code analysts on the Influencer50 Low-Code50 Study for 2019, #5 on Onalytica's list of top Digital Transformation influencers for 2018, and #15 on Jax's list of top DevOps influencers for 2017.

Mr. Bloomberg is the author or coauthor of five books, including *Low-Code for Dummies*, published in October 2019.

About Splunk Inc.

Splunk Inc. (NASDAQ: SPLK) turns data into doing with the Data-to-Everything Platform. Splunk technology is designed to investigate, monitor, and analyze and act on data at any scale.

Image credits: Chris Fithall, Hefin Owen, Ozzy Delaney, Roger Mommaerts, and Tomas Kohl.