Since I joined Akamai as a Solutions Architect six years ago, only one thing about streaming has changed: Everything.

Early in my tenure, online streaming experienced a significant milestone: the iPhone 5 launch keynote. I got my first hard lesson in delivery while watching the stream break up in real time. Everyone had underestimated how popular the event would be.

The issue turned out to have had nothing to do with the stream. It was an API call that went bad – the entire website was relying on one API, which put too much load on a server and crashed the site. If there's no site, there's no video.

This is the kind of painful learning experience we all encounter as we navigate the complex ecosystem of large-scale events.

Since then, streaming has evolved significantly. And just as we get comfortable with the new normal, it too will change. But within the continuous change, there are a few constants. Here are some of the strategies I’ve learned to use to make delivering big events a successful, positive process.

1. Don’t underestimate your blind spots

When you work on something as rapidly evolving as streaming, it’s pointless to expect perfection from your workflow, technology, or team. Despite your best efforts, things will go wrong. Your backup plan does not back you up unless every component in the solution is aligned.

From my vantage point, success lies in focusing on the linkages. Yes, map out the components of your workflow. But also map out the processes between people, between technologies, and between people and technologies.

Make sure all key stakeholders have eyes on those workflows. Give smart people visibility into those workflows, maps, and processes, and make it safe to ask questions. You’ll be amazed how many blind spots you will find.
2. Test everything

Sometimes, however, even proactively searching for blind spots isn’t enough. I know of one broadcaster that obtained rights to the newest season of a popular series. It properly focused on delivering that season with the best quality possible. But because the broadcaster had no previous experience with “must-watch” TV, it hadn’t anticipated the surge of catch-up viewers hitting previous seasons, and its workflow was unable to scale. Many viewers had a suboptimal experience.

How do you protect yourself against what you just don’t know? Testing.

Good engineering and DevOps teams do a ton of testing, whether the event is a streaming or non-streaming event. They put individual components through rigorous quality assurance (QA) processes.

But not all teams test end-to-end workflow – how individual components work with each other – for each of the most common viewer journeys, from sign-up to payment to viewing. Few teams test these viewer journeys in production, and even fewer test them at scale.

These last tests are critical. Unlike cacheable data like video, dynamic events like login or entitlement look-ups need production testing at greater-than-estimated scale to truly see where the break/fixes occur.

At the same time, all it takes is just a few milliseconds for one media segment to be late in the live streaming workflow for a user to have a flawed experience.

If the broadcaster I mentioned above had followed this kind of protocol, the stream might not have crashed during catch-up season.

3. Win as a team

Video delivery is not a game of King of the Hill. If good workflow needs good choreography, so do the teams that use this workflow. And there are a lot of teams involved: rights holders, identity providers, payment providers, authentication service providers, security partners, and so on. When these teams work in isolation, invisible holes can open.

This happened several years ago to one of two providers that had each acquired rights to a major sports event. One approached delivery as a shared effort across all the stakeholders to “make the dream happen.” The other worked with vendors in isolation from one another. That provider was unable to deliver because
of a small technical hiccup that might have been caught had everyone shared insight into the workflow. The company took a financial and reputational hit, and some individuals lost their jobs.

Visibility

One of the important benefits of a robust testing environment is how it boosts the confidence of the teams who are delivering the event and in the resilience of your workflow.

I’ve been in great testing situations where a light bulb goes on for someone: “What if…?” We can then test against this idea, enriching our delivery, and building our confidence in our ability to deliver.

There is no such thing as “too much data” when it comes to events. You don’t know what you don’t know, so it’s crucial to have visibility into every component of the workflow, in as much detail as the data allows. Consider visibility a key criterion as you start planning such events. You never know when and how 2018 data might be useful for an event in 2020.

Keeping It Real

We’ve all been in war rooms when stuff starts to go wrong. When you can replace stress with confidence, you have more capacity to think clearly and make the quick decisions you need to make when the event goes live.

By staying real about your blind spots, developing a comprehensive testing regime, and building a cross-functional team supported by great testing capacity, you have the makings of a successful large-scale online event. And that sure beats panic.