→ PLAYBOOK

Deploying interactive 3D wherever your target audiences are

How to deliver your creations at scale: A playbook for device-aware deployment with the right fidelity, footprint and per-platform rollout strategy.

Unity®

Courtesy of Siemens

Courtesy of Esri


Courtesy of Random42

Introduction

No matter how polished or powerful a real-time 3D experience is, its usefulness depends on who can access it and where. Once you've connected and standardized your 3D asset library and optimized your models for their target use cases, the final step is to make those experiences available where they need to be — across desktop, mobile, web, XR headsets and HMI systems. "The most common platforms, like desktop and mobile, are naturally the easiest to deploy to," said Henning Linn, senior director, industry customer success, at Unity. "But XR and VR can be a bit tricky, because form factors aren't very heterogenous. HMI is even more complicated because every OEM does it differently."

Whether you're hosting an in-depth training walk-through using an XR headset, collaborating on design plans or delivering engaging customer experiences, each platform has its own strengths, weaknesses and use cases. The web is great for accessibility, XR or VR is the go-to choice for immersion, desktop and mobile are suited for everyday utility, and HMI delivers embedded intelligence. In a typical modern industrial setting, you'll probably use a combination of these platform types.

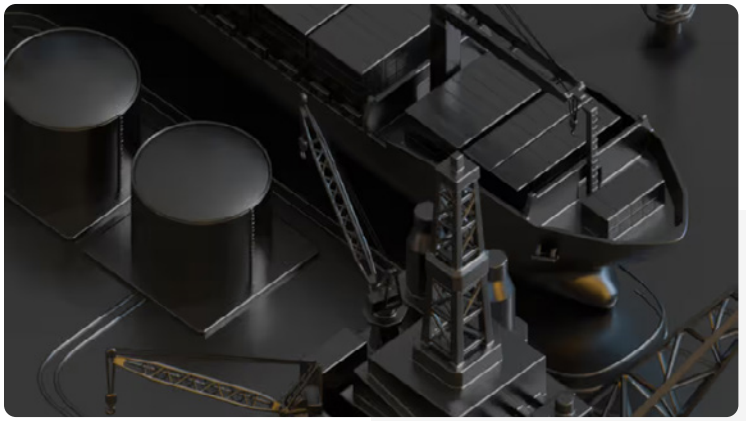
The goal now is to make those designs available to those who need them. But that's not as simple as it may sound because of the immense diversity of devices, each of which has its own capabilities and limits. That's especially true when it comes to customer experience (CX) use cases, where you need to accommodate the broadest range of platforms and devices. However, even internally, for use cases such as training, design and engineering, every team has different needs, expectations and devices.

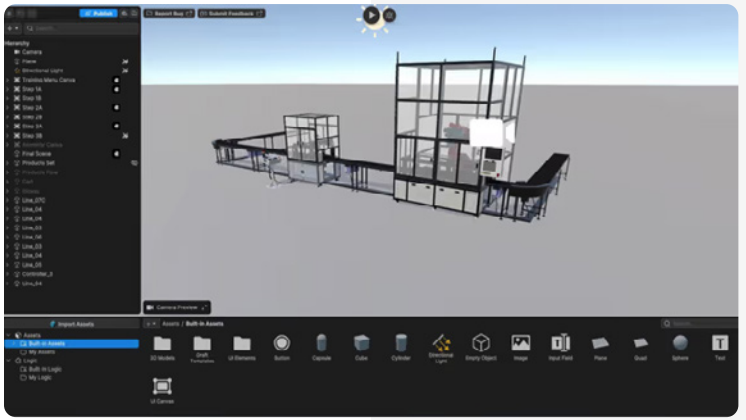
This playbook, the third and final in our series, assumes you have a centralized, connected 3D asset library and you've built and optimized the experiences for your target use cases. Here, we'll look at how to deliver those experiences to your stakeholders or customers no matter where they are or the devices they use. Deployment is all about removing that last mile of friction between great experiences and the people who'll benefit from them.

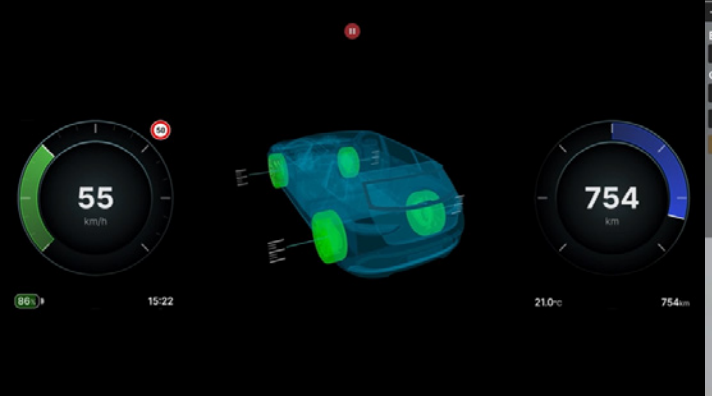
The last mile: Closing the gap between the lab and the field

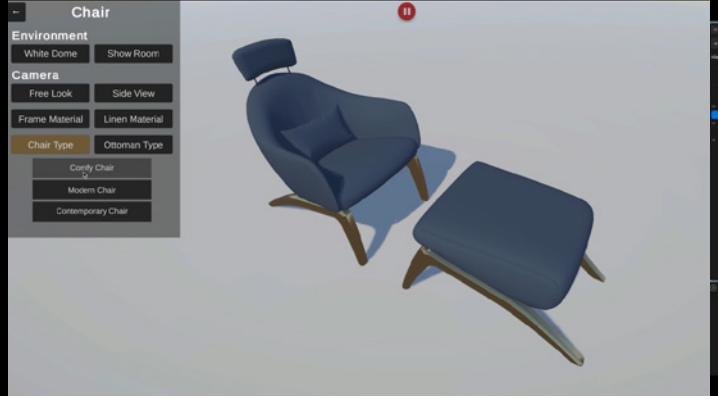
You've probably seen it before: An interactive 3D product demo looks great on a high-end workstation, but the moment it ends up on a midtier tablet on the factory floor, it falters. Frame rates plummet, downloads drag on, XR-tracking stutters under harsh lighting, and the pilot stalls before stakeholders ever feel the value. The overarching theme here isn't that the experiences are necessarily "wrong" — they're just not being deployed in the right way in the right places.

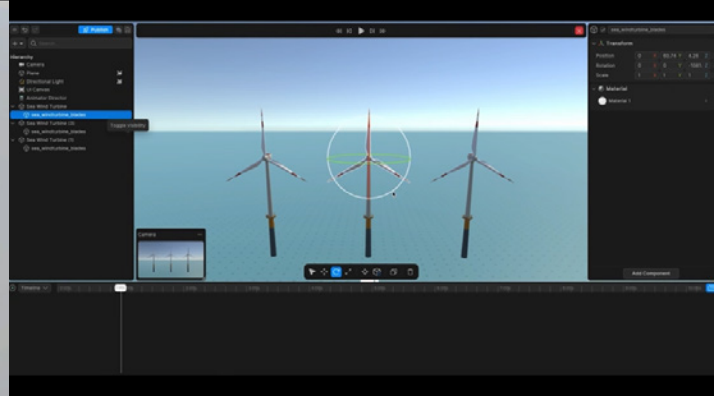
"For many organizations, the issue is a limited pool of developer resources or access to funding, so teams often get stuck at the proof-of-concept stage," said Sarah Lash, senior vice president and general manager, industry, at Unity. "If that happens, the ability to iterate or make changes is limited, which is why we're working on our Unity Studio solution that allows nontechnical teams to whiteboard in a Unity application and start that design and iteration process themselves."











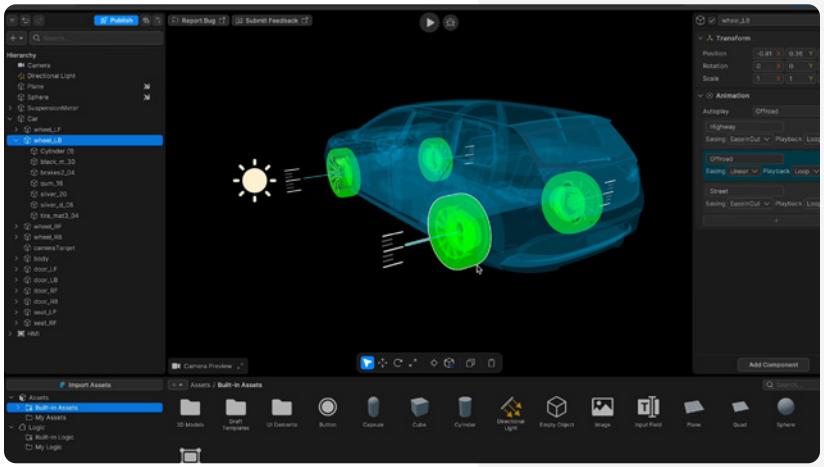
From a technical perspective, there are several common reasons for deployments to fail:

- Performance mismatches happen when an experience isn't optimized for the target hardware. It's normal to use high-end workstations to build and prototype experiences, but you also need to adjust for weaker platforms, such as mobile, web, smartphone or XR.
- Device compatibility and variation result in experiences that might work well on one device but not on another. Experiences can even differ greatly between systems of the same type, such as different web browsers, smartphones or graphics processing units (GPUs).
- Varying real-world conditions can also get in the way of great experiences. For example, an XR application might work in a typical lab environment, but the moment it ends up on a brightly lit factory floor, motion tracking could become unstable if not suitably optimized beforehand.
- Scalability problems, especially in the case of larger projects, tend to manifest because of a lack of robust multiplatform or multiuser architecture. For instance, a content delivery network (CDN) might not be ready to handle multiple simultaneous downloads of a large model.
- Finally, there's the human element, where a lack of sufficient training or support for end users can lead to low adoption rates. Even something seemingly minor can result in users seeking undesirable workarounds or, in the case of customers, turning to competitors instead.

Of course, some devices are more challenging to accommodate than others, but again, it depends on the use case and your desired outcome. Web browsers, despite being universal, are one of the tougher platforms to accommodate because of limited memory and a lack of a dedicated GPU in many cases. XR headsets (AR or VR) present challenges in both performance and comfort. Embedded HMI systems, such as those found in vehicles or on assembly lines, tend to run on highly customized hardware, which is rarely best-in-class when it comes to graphics processing.

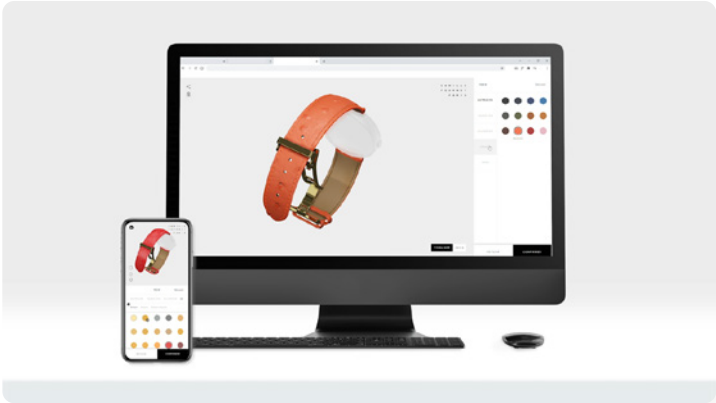
The list goes on, but there is a foolproof way to test deployment readiness and optimize as necessary. By testing on a range of target devices in the appropriate real-world conditions, as early and as often as possible, you can iron out any issues before it goes live. “The earlier you have a chance to test things, the easier the final production deployment will be,” Linn said. “Test on real end points earlier so deployment ‘just works’ when it matters. For example, start with complex device types like embedded HMI or XR to better understand the hardware patterns and requirements and iterate from there.”

> “The earlier you have a chance to test things, the easier the final production deployment will be. Test on real end points earlier so deployment ‘just works’ when it matters.”
>
> Henning Linn
> Senior Director, Industry Customer Success, Unity



Optimize for performance across devices and end users

Deployment isn't about rebuilding your asset library or redesigning your creations. It's about choosing and accommodating your target devices and optimizing the tool chains that connect experiences to the desired outcomes. A single authoritative 3D model can — and should — work well on any platform, provided you plan for multiple levels of detail (LODs), informed lighting choices, and adjustable physics settings tied to device types and their capabilities. As Linn said: "It really depends on what you want to achieve, but you can come to some good results with certain compromises, and that applies to HMI and XR experiences, too. It's important to know all your end points up front and ramp up certain fidelity levels for higher-end end points."



Courtesy of SmartPixels

Courtesy of Prespective

1. WEB:
Universal low-friction access

Nothing beats the web for reach and accessibility. Most web browsers are largely cross-compatible in that they all support HTML5, CSS and JavaScript. At the same time, even modern browsers share much the same limits, especially with regards to download sizes, memory footprints and startup times. However, rendering 3D graphics without plug-ins — which is necessary for supporting the vast majority of users — also requires adequate hardware. For example, while integrated GPUs can handle lighter scenes, higher payloads and shader complexity will bog down most midtier laptops or mobile devices.

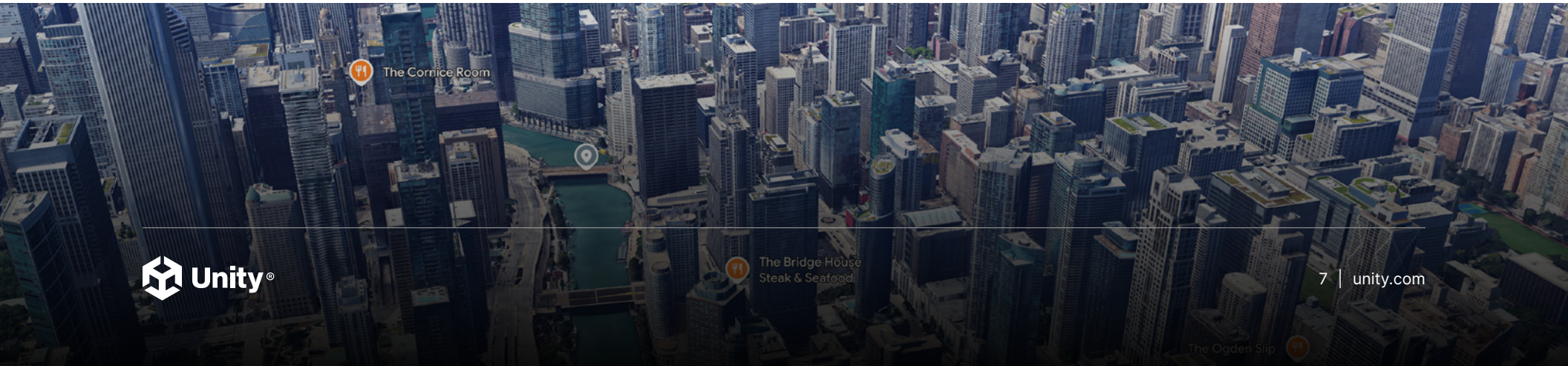
Treat initial load times as prime real estate, especially in the case of customer-facing experiences. If an experience doesn't load up in seconds on a typical broadband connection, people aren't likely to stick around. For more complex assets, it's best to stream multiple LODs on demand, rather than trying to load everything at once. Also, keep an eye on device variability.

Older laptops with integrated graphics, for instance, will expose edge cases where the experience stutters or lags. To avoid these pitfalls, test your experiences on multiple midtier devices, set fallbacks for unsupported features and aim for at least 30 frames per second (FPS) for casual viewing.

WHAT TO AIM FOR →

Aggressive control over texture size with multiple LODs, removed or simplified meshes for invisible internals, and streamed content loading that prioritizes "hero" surfaces (critical areas of detail — such as glass, metals, paint or fabrics) and interactions.

Courtesy of Google Maps



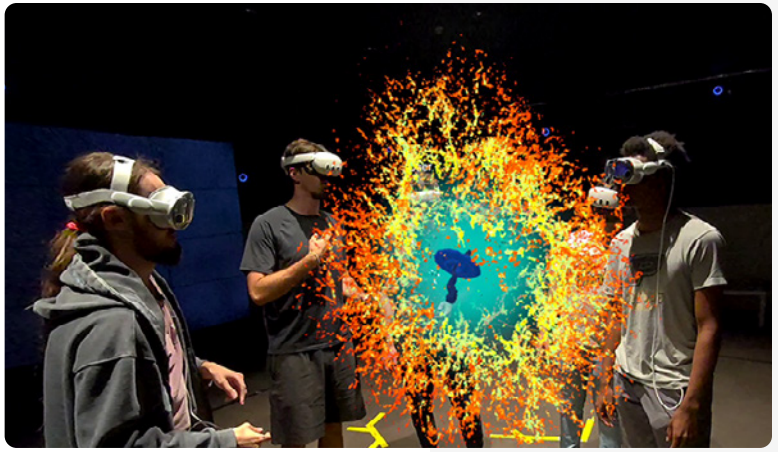
2. XR:
Immersion that demands stability

Extended reality experiences, such as virtual reality (VR) and augmented reality (AR), are unbeatable in cases where spatial understanding is important, such as hands-on training and complex walk-throughs. On the other hand, they're also the least forgiving of performance lags. Most stand-alone headsets use mobile-grade chipsets, albeit relatively high-end ones, but they require high stable frame rates to avoid user discomfort and motion sickness. As a general rule, the experience should render at the device's native refresh rate, which, in the case of VR headsets, tends to be 90 or 120Hz, equivalent to the same in FPS. Consider 72 FPS the bare minimum for comfortable VR headset experiences.

AR requirements vary greatly depending on the target device. AR inherits much of the same comfort demands as VR, so it's important to aim for stable frame rates of at least 72 FPS. On phones or tablets, AR incorporates camera feeds and simultaneous localization and mapping (SLAM) to the render loop, which, on modern devices, means aiming for 60 FPS, with a strict minimum of 30 FPS. Outdoor lighting and thermal throttling can still be an issue, however, so plan for adaptiveness with dynamic resolution and readily accessible feature toggles.

WHAT TO AIM FOR →

Use prebaked lighting in most cases, multiple LODs and simplified shaders, especially if you need to accommodate low-end XR headsets or mobile AR experiences.



Courtesy of CollabXR

3. DESKTOP:
High performance for demanding experiences

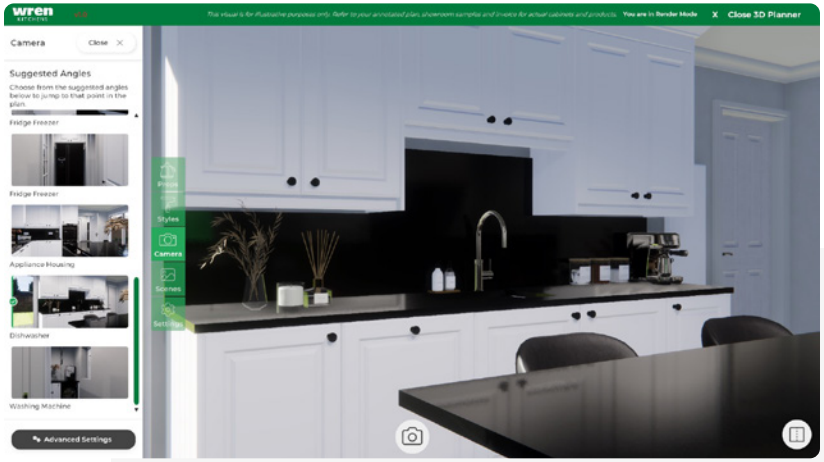
Desktops are ideal for detailed analysis or multimodal reviews, but they also serve a wide range of scenarios that don't necessarily need a flagship GPU. For example, day-to-day design approvals, quality checks, safety walk-throughs or manufacturing planning benefit most from responsive interaction and clear inspection tooling than from maximum fidelity, so target devices will often only be equipped with midrange, discrete GPUs or newer integrated graphics for lightweight viewing. In these cases, prioritize deterministic review tools, such as sectioning (cutting planes to reveal internals), measuring (snap-accurate dimensions) and exploding (staged assembly views with part callouts).

For heavier internal workloads, such as rendering massive assemblies and complex simulations, dedicated workstation-grade GPUs or remote GPU resources are the norm. However, not every end user needs such a costly device. High-end local hardware remains unrivaled for teams that author highly complex content, but for consumers, streamed sessions from a GPU server are far more cost-effective.

This also avoids the "but it runs on my RTX" surprises and reduces support load across mixed fleets of devices. However, network bandwidth can become a bottleneck, so it's important to incorporate local caching so that reviews don't stall if the network can't keep up.

WHAT TO AIM FOR →

Default to the highest stable tier, ensuring consistent frame rates on the target devices. Use deterministic review tools and on-demand streaming, and prioritize input responsiveness.



Courtesy of Wren Kitchens

Courtesy of Icon Group

4. EMBEDDED HMI:
Reliability and instant clarity

HMI brings 3D into the product itself. Think in-vehicle displays, industrial control panels and medical devices, where uptime and predictability are far more important than photorealism. In these use cases, operators often have to make time-critical decisions, while most embedded targets run on constrained hardware with long service life cycles. Even where premium hardware such as dedicated GPUs is available, it's rarely refreshed on consumer timelines; embedded devices on assembly lines and vehicles need to stay stable for years, so software has to do the job of managing tight hardware budgets.

Common use cases include setup, guided maintenance and training on assembly lines or visualizations of arm kinematics, reach envelopes and no-go zones in robotics systems. For automotive use cases, interactive 3D might be used in advanced driver-assistance systems. In these cases, legibility and motion smoothness are nonnegotiable. This is also true of many medical and lab devices, where the bar is also heightened by regulatory and safety demands.

WHAT TO AIM FOR →

Simplified geometry and prebaked lighting significantly reduce performance demands and loading times, while long-term support from vendors ensures lasting support for embedded devices.

Operate and scale your deployments for lasting success

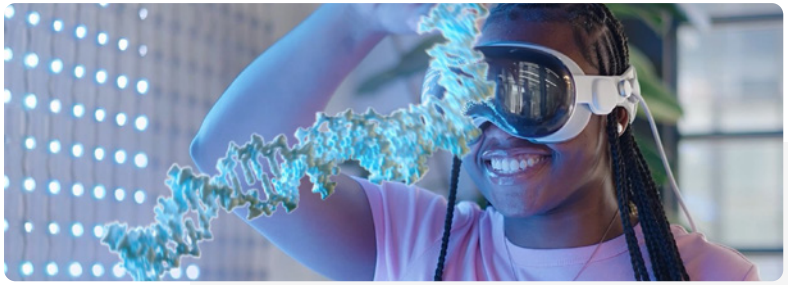
Teams often stumble by designing for the highest-end device, then accommodating lower-end devices by copying and pasting UI elements across interfaces and front-loading performance-hungry assets. However, with the right end-to-end platform, it becomes much easier to avoid these pitfalls. For example, using automatic scaling and streaming heavier assets on demand, experiences can degrade gracefully rather than failing abruptly. This ensures business continuity by providing some predictability for refreshes.

Unity's profiling tools are built to specifically accommodate these challenges: "We monitor things like asset-loading times, startup times and so forth," Linn said. "These are especially important in HMI deployments, where you need quality real-time feedback on how the model behaves on the actual device."

Initial deployment is just the beginning, especially given the dynamic nature of projects such as real-time digital twinning of production lines or shop floors. Operating successfully at scale requires a predictable way to deliver content without version drift, telemetry to see what's working (and what isn't), and governance that lets you move quickly — without breaking things.

"As for long-term rollout strategies, I think it's much the same as with any software, where you scale in waves and target the end points and users who can give you the most valuable insights on usage to identify any bugs or missing features early in the process, before you roll out to the next user or device group," Linn said. "Eventually, you'll reach full general access, by which time you'll have minimized the bugs and maximized the quality of the user experience."

That means paying close attention to update cycles, device telemetry, governance and long-term support. "Track things like session length and adoption rates, and seek direct end-user feedback where possible on things like the quality of the experience," Linn recommended.



Courtesy of Random42

Content delivery and rollout

Phased rollouts tend to work best, especially when you're targeting hundreds or even thousands of end users. With large-scale projects, such as customer-facing experiences or those spanning multiple factories and facilities, start with a single site or target user group, monitor for issues, and scale from there. This mitigates the effects of unforeseen problems and makes support much more manageable. You might also use remote configuration to ship the application or experience to everyone, but only turn specific features on for selected locations or user groups. Plan the network side, too, by precaching larger models using a suitable CDN, and schedule any major updates or downloads off-peak to prevent the first day becoming a bottleneck.

Device data that matters

You won't be able to tell if an experience is delivering the outcomes you're hoping for if you're not collecting the right telemetry data. To answer the critical questions of who's using it or whether it's working as intended, you'll need to track adoption and engagement metrics. For example, tracking session lengths and how often people come back gives you a broad idea of the project's success. However, deeper metrics, such as those spanning quality and performance, are important for continuous improvement and integration. For specific workflows, such as training simulations, define simple workflow steps to show the percentage of users starting a training step compared with how many finish it. Finally, break things down by device type to see if any group needs extra attention.

Courtesy of Vodafone Deutschland

Courtesy of Visionaries 777

Governance without complexity

Governance involves setting clear rules for who manages updates and how rollouts are organized to mitigate disruption and risk. Ideally, each release should have one named approver and a backup, with all decisions being logged in a shared place. You'll also need checks before anything goes live, but try not to overcomplicate things. For example, performance checks on target devices and sign-off from product and operations teams is usually enough, with the exception of use cases involving regulated or sensitive data, which will also require compliance and security reviews. During the release window, including the release of any updates, make sure to have a support owner on point with clear escalation paths so issues can be resolved quickly.

Long-term and community support

Hardware, drivers and operations never sit still, so always plan for long-term stability and adapt your deployment strategy to accommodate new platforms and updates as and when they're released. Working with specialized industry vendors that offer long-term support commitments also ensures you can continue supporting older devices, which is especially important in the case of high-end industrial systems equipped with embedded HMI that can't easily be upgraded. Many vendors offer open-source tools and assets as well, in which case you can treat the surrounding community as part of your support system by reusing proven assets, tracking release and update notes, and maintaining an internal knowledge base of best-in-class drivers, tools and device and use case-specific guidance.

Getting started: Your step-by-step deployment plan

Deploying interactive 3D experiences where your stakeholders and customers are allows you to deliver another dimension of information on any device. Whether you're hosting a cutting-edge training simulation, collaborating on design across desktop and mobile, or building customer experiences for web browsers, choosing the right platform ensures you get optimized results for any target.

Here's a quick recap of what it takes to get there:

1. Map stakeholders to devices and user contexts.
2. Define performance benchmarks per platform.
3. Pilot deployment across a limited number of platforms to surface issues early.
4. Set up content delivery and telemetry.
5. Establish governance roles for updates and support.

Are you ready to connect, create and deploy immersive experiences for your industry? Unity Industry provides a comprehensive suite of tools for training and guidance, 3D design collaboration, CX, XR, and HMI backed by three-year long-term support and a massive developer community.

GET STARTED →

About Unity

Unity [NYSE: U] offers a suite of tools to develop, deploy, and grow games and interactive experiences across all major platforms from mobile, PC, and console, to extended reality. For more information, visit unity.com

LEARN MORE →