info@apriorit.com
www.apriorit.com

Case study

Developing Drivers for Low Latency Virtual Reality Headsets

Our client is a manufacturer of virtual reality (VR) equipment used for industrial purposes. They wanted to create a new device and needed a team with expert driver development skills to develop the drivers for it.

The Apriorit team successfully delivered drivers that enable fast data transmission with low latency. We also helped our client reduce the time to market for their device by developing and testing the drivers in parallel with development of the device itself.

The client

Our client develops VR headsets designed for professional training. Among other use cases, these devices help doctors train to perform surgeries, scientists train to conduct experiments, and pilots train to navigate aircraft.

To support such critical use cases, our client has to ensure the VR experience is as realistic as possible. Our client's devices deliver the best possible video quality, seamless user interactions, and the lowest possible latency for video delivery. To achieve this, our client is constantly improving device hardware and needed a highly skilled development team to create fitting software.

The challenge

Our client was looking for an expert driver development team that could create VR headset drivers in parallel with their hardware development efforts. They had high requirements for their device drivers:

- Low latency for data transfer from the device to the application
- High bandwidth for data transfer
- Data transmission speed up to 10 Gbps
- Stable performance

Our approach

To deliver drivers that fit our client's needs, we developed the following:

1. A high-speed data transmission protocol for a PCI Express device
2. A Windows device driver for a PCI Express device
3. An API for the Windows device driver

Developing drivers for virtual reality headset without the physical device was tricky because we couldn't test them with the target hardware. To overcome this challenge, we discussed our customer's hardware specifications and used Quick Emulator (QEMU) to configure a virtual device that replicated the VR headset. In this way, we were able to deliver the first version of the drivers before the client finished the first version of the hardware.

After analyzing the client's requirements, we formed a dedicated team and selected the technology stack for this project:

APRIORIT DEDICATED TEAM			
Business analyst	Project manager	Driver developers	QA specialists

KEY TECHNOLOGIES				
C/C++	OpenCV	DirectX	QEMU	Windows Driver Framework

TECHNIQUES WE USED		
Multithreading	Real-time image processing	Driver profiling

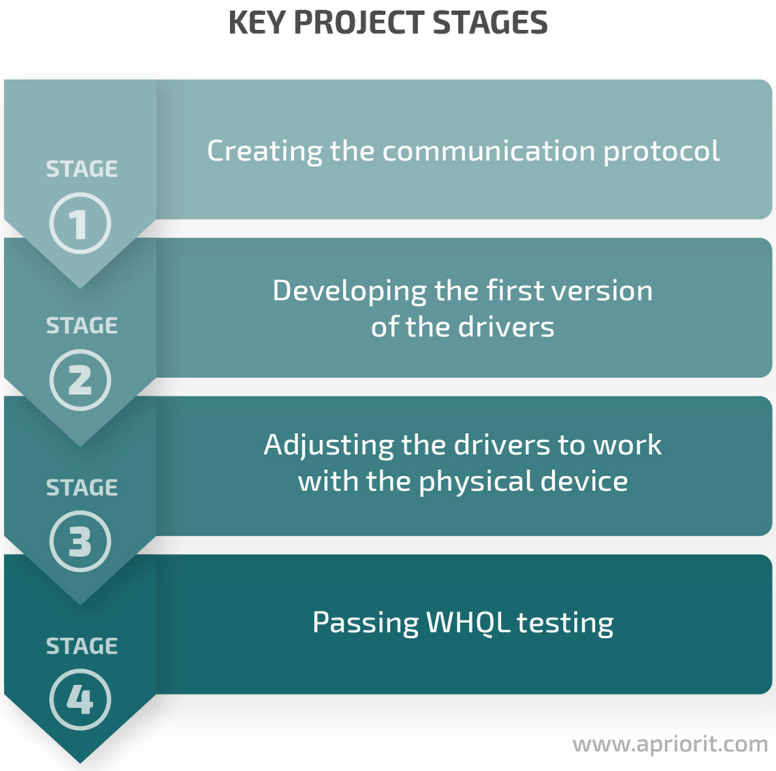
The result

We developed the first version of the drivers before the client even built the device. After the headset was built, we rolled out several iterations of drivers to stabilize the headset's performance and adjust the drivers in response to hardware improvements made by the client. Finally, we tested the drivers according to the Windows Hardware Quality Labs (WHQL) requirements and acquired a WHQL release signature from Microsoft.

With our help, our client successfully released their device on time. They were impressed with our team's performance and hired us to develop drivers for two new devices: the second generation of the VR headset and its portable version.

How we did it

The whole project, from the discussion of the initial software requirements to the release of the device, lasted around one year. During this time, we created project documentation, solved many development and testing challenges, released several iterations of drivers to ensure the best possible performance, and tested the drivers according to WHQL requirements. There were four key stages in this project:



Before developing drivers for VR headset, we had to elicit and collect all the requirements for the software. This was especially important because the hardware and drivers were being developed by two teams on different continents. Despite that, those teams had to work in parallel.

That's why we started the project by creating a communication protocol.

1. Creating the communication protocol

Our communication protocol document described the following:

- Functional and non-functional requirements for device communication
- Functionalities to implement
- Use cases
- Development stages
- Deadlines

This document reflected all communication about the project between our team and the client's team. It helped us formalize all driver requirements, discuss them with the hardware developers, keep track of changes, and ultimately save a lot of time during development.

2. Developing the first version of the drivers

While developing the first version of custom drivers for VR headset, we successfully solved many tricky tasks, including:

- **Reducing the latency.** Low latency was one of our client's top requirements because it allows a headset user to interact with the virtual environment in real time. To achieve low latency, we optimized data delivery mechanisms and assessed the performance using the results of driver profiling and testing.

- **Adding Secure Boot and Device Guard support.** These Windows protocols ensure that the Windows operating system runs trusted versions of applications and device drivers. Also, support for these protocols is required to make drivers fully compatible with Windows 10 and Windows security policies.

- **Synchronizing hardware components.** The hardware consisted of independent devices that needed to be synchronized in Windows. To avoid that, we added support for device synchronization to the driver.

- **Resource allocation for device operation.** Windows didn't always allow the driver to allocate the needed amount of resources upon connection of the device without restarting the operating system. We ensured the allocation of needed resources without a system reboot by changing resource reservation methods.

3. Adjusting the drivers to work with the physical device

When the client finished working on the hardware, we needed to adjust our custom VR device driver to work with it. Thanks to our use of a detailed communication protocol and QEMU prototype, there were few changes we had to make to get the hardware and the drivers to cooperate.

When the device and drivers were working together smoothly, the client started to improve the hardware performance and feature set. We updated the driver software accordingly and used hardware improvements to gradually reduce the latency of video delivery to 3–10 milliseconds. At this stage, we also expanded the VR functionality of the drivers.

To manage the development process at this stage, we implemented the following cyclic workflow after consulting with the client:

- Create a list of features to add and discuss their requirements.
- Add the discussed features to the communication protocol and confirm the protocol with all project participants.
- Develop drivers, the API, and samples as well as cover the drivers with tests.
- Oversee the implementation of new features in the firmware, track bugs, and monitor regression using driver integration tests.

Here are the key tasks we undertook at this stage:

- **Improved the compatibility of the device and driver interfaces.** Under particular configurations, the device and drivers didn't work well together. We fixed the driver interfaces to ensure the software supports any device configurations.
- **Resolved several hardware issues.** We discovered several issues with Intel devices and discussed possible ways to resolve them with the Intel support team. Also, we discovered a bug in the Windows 10 interrupt controller and reported it to Microsoft. To avoid waiting for the fix, we improved our drivers to make sure the bug wouldn't affect their performance.

4. Passing WHQL testing

It was important for our client to ensure the compatibility of their device with Windows 10. The best way to do that was to test the device drivers according to WHQL standards.

WHQL is a series of tests designed by Microsoft for third-party developers creating Windows-compatible hardware and software. These tests help Microsoft determine that hardware and software is compatible with Windows, works securely, and contains no critical issues.

To confirm that our drivers were up to Microsoft's standards, we ran a series of automated stress tests, sent the results to Microsoft for review, and successfully passed them.

Obtaining WHQL certification allowed our client to:

- Add their drivers to the Windows Update service and Microsoft Compatibility Center
- Mark their devices with the official Certified for Windows sticker
- Simplify driver installation for end users

The impact

Our driver development and virtualization expertise alongside a carefully planned and documented development process helped our client successfully release their product, and they received positive feedback on it from their customers. The drivers are WHQL certified and helps the device operate with 3–10 ms of latency and ~11 Gbps data transmission speed.

Inspired by this result, the client is continuing development of the project. Now, they're working on the second generation of the device and have trusted us to create a device driver for the new VR headset.