



## CASE STUDY

# Nearshore Software Development

Custom oilfield automation and PLC monitoring that solves background operation for Field Solutions. Totally user friendly, it makes their client's lives easier allowing quick changes and offering a qualified member attention at any time. Their background is purely oilfield automation.

Founded in the heart of the Permian Basin, **Permian Controls'** systems have been honed in some of the harshest operating conditions the oilfield can provide. They're committed to serving the hard-working producers of the region and across the country.



### Nearshore Solutions

**Industrials**

### Development Expertise

**ASP.NET**

**.Net**

**PostgreSQL**

**SCADA Experienced**

### Outsourced Services

**Project Management**

**Dedicated Team**



*"Azumo is one of those rare nearshore software development companies that understands the intersection of technology, its relation to business enablement and client experience. We couldn't be happier with our choice of a partner and custom software developer."*

## THE CHALLENGE



Permian Controls had a legacy system that supported a large number of revenue-generating customers.

The system while functional was prone to security breaches and was difficult to maintain and integrate to new-generation apps.

They needed help to migrate the solution to a more modern, stable, and resilient codebase, without disrupting the customer base. We took the challenge and went far beyond the expected results...



## CASE STUDY

## THE SOLUTION

We built a dedicated team consisting of an architect, two senior developers with DevOps expertise and a project manager. Given the age of the existing solution, there were few skilled developers available who understood the requirements of the backend, the solution and the modern applications to communicate with – so we got them.

We migrated the solution completely from an outdated codebase into a modern, maintainable and resilient solution, with no downtime, while simultaneously supporting the business' customer expansion. As the business grew adding new customers, the custom PLC programs we created supported the additional needs.

We also created live dashboards using LogicMonitor for both live and historical queries/reports as well as alarms and compliance management. Finally, our design had the additional benefit of significantly reducing the computing costs – as we moved the solution from an on-premise implementation to the cloud-, reducing the all in cost by 90%, even as the cluster tripled in size.

### Dedicated Team with Experience in the Oil Patch

We built a dedicated team consisting of an architect, two senior developers with DevOps expertise and a project manager. Given the age of the existing solution, there were few skilled developers available who understood the requirements of the backend, the solution and the modern applications to communicate with – so we got them.

### Modernize the codebase

We migrated the solution completely from an outdated codebase into a modern, maintainable and resilient solution, with no downtime, while simultaneously supporting the business' customer expansion. As the business grew adding new customers, the custom PLC programs we created supported the additional needs.

### Connect to LogicMonitor

We also created live dashboards using LogicMonitor for both live and historical queries/reports as well as alarms and compliance management.

### Total Solution Reduced Cost

Finally, our design had the additional benefit of significantly reducing the computing costs – as we moved the solution from an on-premise implementation to the cloud-, reducing the all in cost by 90%, even as the cluster tripled in size.

## THE RESULTS

Azumo maintained a number of VisualBasic daemons that listened to PLCs in the field and captured data from them to store on a SQL Server database. We also maintained an ASP site that allowed the data to be queried and reports to be created.

The site also triggered alarms for Oil Rig personnel whenever appropriate. In parallel, Azumo progressively replaced the daemons and front end.

We created one-to-one (1:1) daemon replacements using Nodejs and MongoDB. This approach enabled us to run the daemon replacements as microservices running on Linux and set up auto-scaling clusters in AWS.

