Case study

Supporting and Improving Legacy Data Management Software

Our client creates data management solutions for content-critical businesses. While they were working on a new cloud-based platform, they needed an experienced team of backend software developers with data management skills to support their legacy software. The Apriorit team has a lot of experience developing and supporting custom data management systems, so we were glad to help.

During this project, we improved client's legacy software for data management and end users' satisfaction with it, implemented new features for better supportability, and helped the client prepare users for migration to a new platform.

The client

Our client is a US-based company that develops solutions for storing, sharing, and transferring data. They have more than 17,000 customers around the world that work in the media, construction, financial, life sciences, recreation, and other industries. For these customers, the ability to manage data securely and efficiently is crucial.

The challenge

When the client approached us, they were supporting a containerized Linux-based legacy solution for synchronizing data between local and cloud file storage. However, they wanted to focus on developing a new cloud-based platform.

That's why they were looking for an outsourced development team that could take over maintenance of the legacy system and solve several legacy system modernization challenges:

- Integrate the team into the client's workflow and infrastructure

- Reduce the number of escalated tickets from end users

- Develop custom features requested by end users

- Improve software supportability

Project challenges

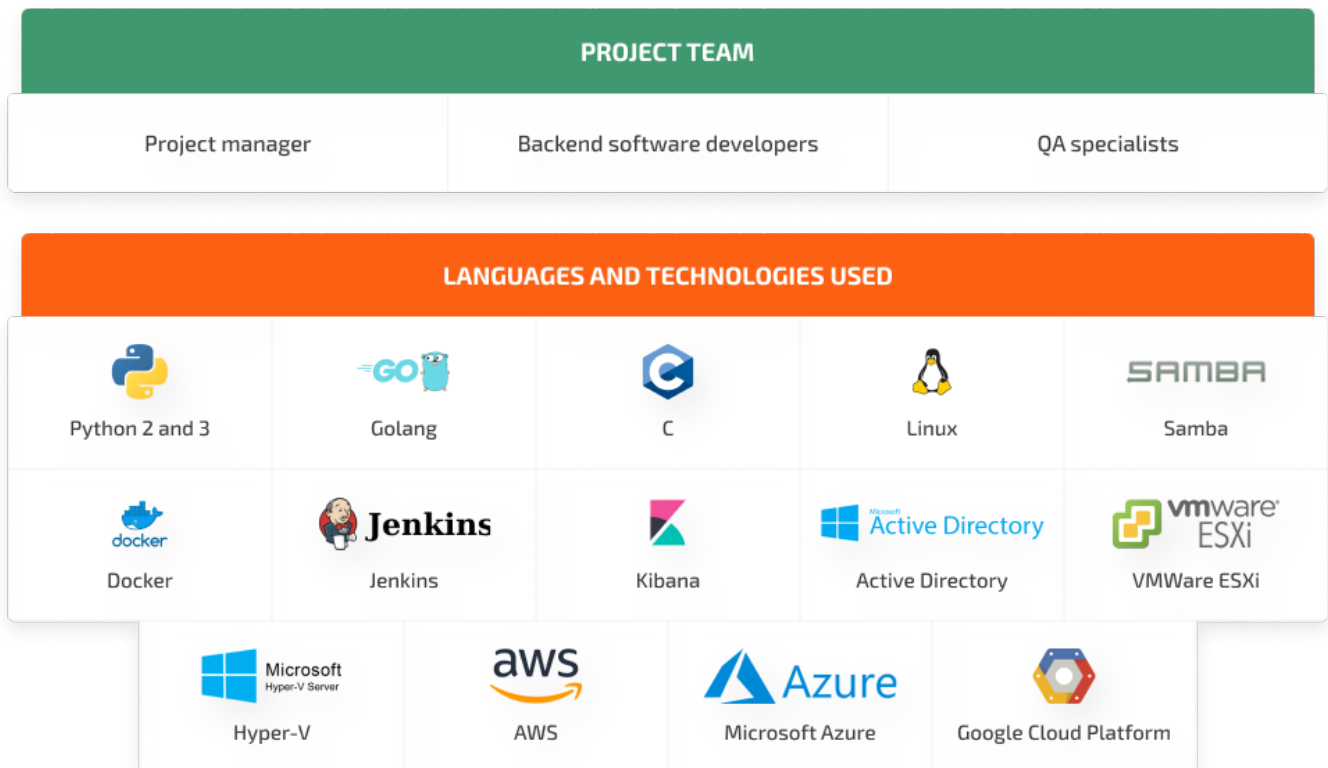| Integrate the team into the client's workflow and infrastructure	Reduce the number of escalated tickets from end users
Develop custom features requested by end users	Improve software supportability

Our approach

To solve the challenges of this project, we formed a team that included a project manager (PM), several expert backend software developers, and QA specialists.

PROJECT TEAM		
Project manager	Backend software developers	QA specialists

LANGUAGES AND TECHNOLOGIES USED

Python 2 and 3	Golang	C	Linux	Samba
Docker	Jenkins	Kibana	Active Directory	VMWare ESXi
Hyper-V	AWS	Microsoft Azure	Google Cloud Platform	

Our project manager (PM) ensured that our development team worked in sync with the client's teams, which were distributed across several countries. The PM implemented elements of Scrum and Kanban methodologies in the workflow and organized two-week sprints and weekly meetings with the client's managers and other team members. This approach improved the project's visibility for our customer, kept the client's team up to date on our activities, and helped us prioritize tasks within sprints.

Our team works hand in hand with the clients development and support specialists in order to improve user satisfaction with the legacy software.

1. We started with refactoring the legacy code and revamping the existing documentation.

2. We also handled tickets escalated by users by fixing challenging bugs and implementing custom features.

3. Finally, we improved the supportability and flexibility of the software by porting it to a new

   operating system, adding new features, and testing new frameworks.
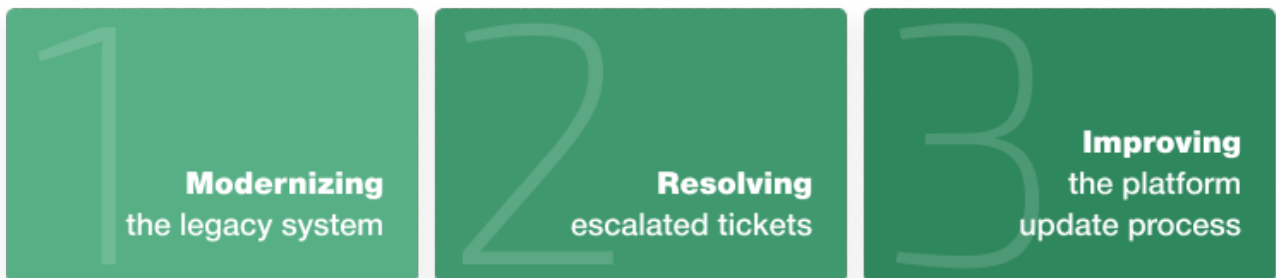
The result

During our cooperation, we helped the client keep end users satisfied with the legacy software by fixing issues reported by end users and developing custom features that users requested. As a result, we reduced the number of escalated tickets and reached the fewest user complaints ever for this project.

Our support for this software allowed the client to focus on building a new cloud-based platform, since their developers didn't have to spend time on bug fixes and legacy software system modernization. Finally, our client rolled out a new product and ensured smooth migration to it. Since they were satisfied with our services, we continued our partnership after the release.

How we did it

During this project, our main tasks were fixing bugs in the legacy system and processing escalated tickets — issues reported by end users that the client's support team was unable to resolve.

Our activities on this project

1 Modernizing the legacy system

2 Resolving escalated tickets

3 Improving the platform update process

Modernizing the legacy software

Working with our customer's outdated system was challenging because of the great amount of undocumented legacy code. Over the years, the software had been developed by different teams using three languages: Python, Golang, and C.

We needed to sort out all the legacy code to fix current issues and simplify future support activities. To do that, we:

- refactored legacy code

- ported core software from legacy operating system to new operating system

- migrated components from Python 2 to Python 3

- added security patches

- reworked old functionalities

- improved the documentation

Thanks to refactoring, we improved the software's performance, reliability, and supportability and resolved numerous escalated tickets.

Resolving escalated tickets

We had to handle two types of escalated tickets: requests to fix existing functionality and requests to develop new functionality.

To solve existing issues, we refactored and reworked code. By the end of the project, we had reduced the number of escalation tickets to the lowest rate ever for this software.

Some end users requested adding custom features to the product in order to simplify their work. We implemented many custom features, including:

- Simple Network Management Protocol (SNMP) integration. The SNMP protocol enables agents to send information about the system to third-party monitoring services. This way, users don't need to log in to the software's Web UI to check the status of and manage each device individually.

- Deselected folder management — This feature automatically manages space on the users devices. When the user no longer wants to access data synchronized to their on-premise device, the software deletes it after ensuring that all updates are saved to the cloud and there is no data loss. It also helps our client save cloud storage space and reduce network traffic.

Though we were working on legacy software, fixing bugs and implementing new features called for frequent software updates. Since updating this software was a time-consuming process, we discussed several improvements with the client.

Improving the platform update process

We needed to improve the software's supportability to help end users with their requests and simplify maintenance. That's why we implemented the following changes:

- **Improved software flexibility with a new operating system.** The software was initially based on CentOS, a Linux-based operating system that didn't afford flexibility. CentOS made patching vulnerabilities and updating software challenging, not to mention that CentOS support will end in 2021. For the sake of rolling out updated and new features, we discussed the need for a more flexible operating system with the client. In the end, we ported the software to Debian and implemented a feature for migrating data with little involvement of end users.

- **Added several update channels.** Initially, the client's software had only one update channel. Adding two more allowed us to control which end users get which updates and gradually roll out new product versions. We added support for three update modes: old stable, stable, and beta.

- **Automated the testing process.** Before our client contacted us, their QA specialists tested any code for their legacy software manually, which was time-consuming. We implemented two automated frameworks, one for regression testing and one for upgrade testing. The first was based on a ready-to-use framework. For upgrade testing, we created a new framework from scratch. These frameworks helped us greatly speed up the testing process by automating certain tests.

The impact

With the help of Apriorit's [data management](#) and [processing](#) specialists, our client ensured efficient support of their legacy software while concentrating on developing a new platform. We put all our efforts into making sure that end users had a flawless user experience when working with the legacy solution.

As a result of our cooperation, our client was able to smoothly migrate to their new platform while keeping all end users.

Working on your own data management software? Benefit from our experience and skills!