

Managed Services to Develop an Automation Framework

“Our [software deployments] before Eliassen Group stepped in were taking 10+ hours, and I had the expectation they’d be able to cut that time down to 15 minutes – they got it down to 2 minutes. Throughout the project, it was clear that their technical subject matter experts (SMEs) were experienced in enterprise initiatives, and their business partners created a seamless completion of the project.”

— **Senior Director, Systems Engineering,**
Telecommunications Company

The Challenge

The client’s challenge involved both automation and migration. They wanted to develop an automation framework to migrate an enterprise legacy application, and they wanted to cut their migration time from 10+ hours to a scant 15 minutes. Alongside this project, the client required a performance testing instance to automate their testing environment, which had the long-term goal of automating their traffic shaping and thereby improving network performance.

The Solution

The client chose Eliassen Group to address this challenge because of our history of automating legacy application processes in enterprise environments. With our guidance, three systems engineers with a focus in automation spearheaded this initiative. To speed up the migration time, the team built an automation framework in Python. To build out the performance testing environment, they leveraged both Python and Locust.

The Client

Telecommunications Company

The client provides residential subscribers with connectivity services, along with broadband for both small business owners and enterprise clients. As of 2020, the company reached 30 million customers, and they are the fastest-growing TV, internet, and voice provider in the nation.

Highlighted Results

- Team cut migration time for an enterprise legacy application from 10+ hours to under 3 minutes
- Delivery of 10 specific performance cases
- Team finished the project 2 weeks early, saving the client 37%

The Result

The team went above and beyond expectations in all ways. First, the client wanted to cut their migration time from 10+ hours to 15 minutes. We cut it down to under 3 minutes. Then we built out 10 specific performance cases so the client could validate the performance and health of their environment. Beyond meeting the targets given to us, our deployment saved the client 37% on the project, and we finished the project 2 weeks early.

Tools Used

- Ansible
- Git
- Locust
- Oracle
- Python
- Shell Scripts