

# AUTOMATING CAPTATION AMOUNT BALANCING THROUGH DYNAMIC SCHEDULED JOBS

## OVERVIEW

*Tetrasoft empowers one of the leading US based health care client in leveraging their vast raw data and transforming it into valuable asset. Our Data Mart application is one such component in the application's architecture which provides insights into the complex provider data extracted from the EDW. This component supports their business in making pivotal decisions on the provider front. Time is critical in the decision making process. We proactively identified the means of reducing delivery time. Saving time in our delivery would help our client in making faster and more efficient decisions.*



## BUSINESS PROBLEM

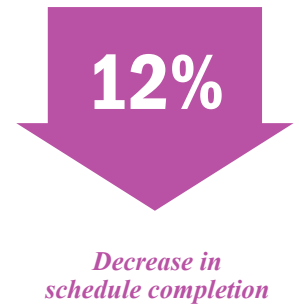
### *“Inconsistency in Reporting”*

Amount attributes in the Data Mart play critical role in decision making of provider's performance. Having large volume of data in the warehouse makes testing these attributes within a stipulated timeframe a challenging one. Adding to the volume of the data, there is a dependency on a manual tester which leads to inconsistency for reporting. Manual testing basically involves a large testing cycle considering the frequency of run for the application. We have set of scripts that would be executed by the tester and would take approximately 8 hours for completion. Until then other jobs would be on hold (due to dependency). Thus automation process seemed a very good choice but a challenging one. Some of the challenges are highlighted below.

The modular approach of automation seemed very out of place here taking into account the multiple applications the attributes are involved in, as this might require generating test scripts for each application which would result in huge effort, less generic and less reusable scripts, and of course more maintenance over the life cycle. The feasibility study revealed that these attributes had a lot of dependencies on other tables.

## ALTERNATIVE SOLUTION

Our focus was to generate very generic test scripts that could be used across different applications and stages in our data ecosystem; the attribute rule was different across applications but followed a pattern. The most important requirement for the framework was to have an ease of understanding and a highly informative report to be generated at the end of test execution, thus allowing those who do not have exposure to the test automation tool to use the framework and interpret the results of the test execution.

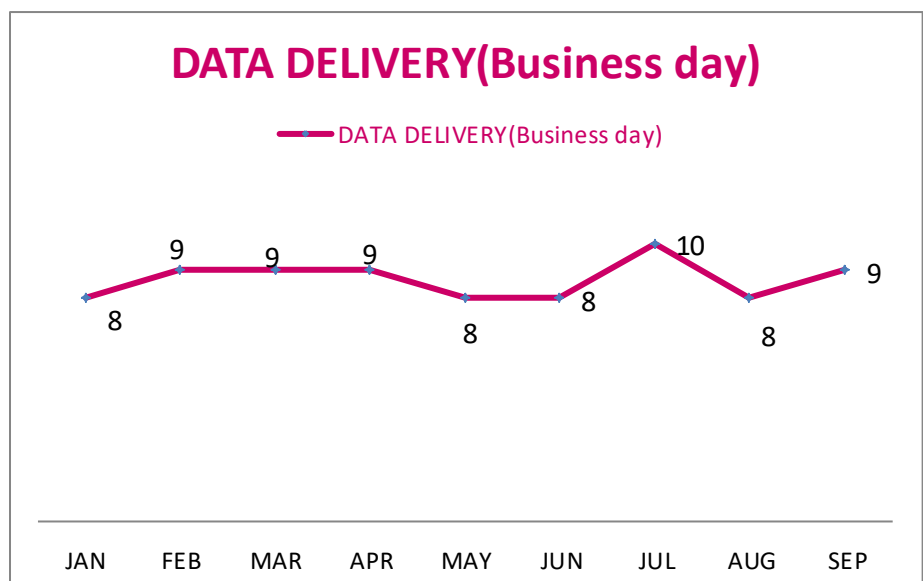


*“Saving 75% of testing efforts each month”*

## IMPLEMENTATION

We created a job and scheduled it to run based on its predecessor. This job would take only 10 minutes to complete all mandatory checks along with amount balancing. There by reducing the manual effort and saving 75% of testing efforts each month.

*“Inconsistent delivery time due to dependency”*

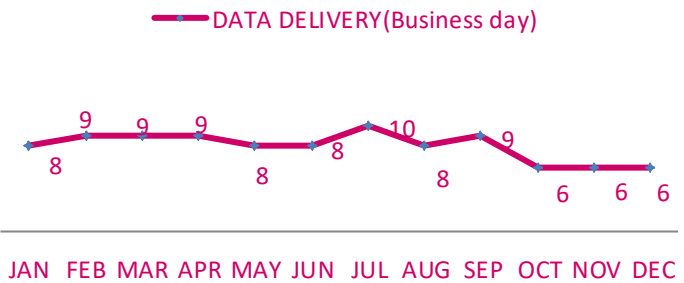




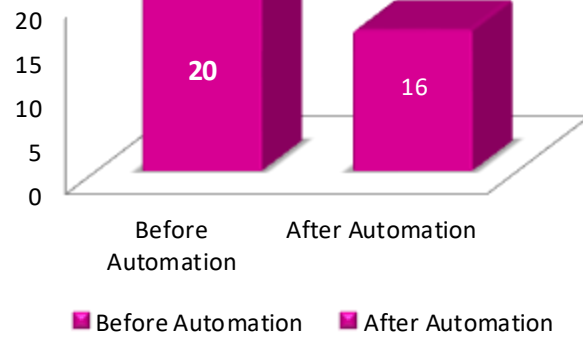
## CLIENT BENEFITS

- **Reduced testing efforts by 75%**
- **Improved schedule completion time by 12%.**
- Report delivery to the business 2 days prior to the SLA thus giving them more time to make decisions through *reliable data*.

## DATA DELIVERY(Business day)



CONSISTENTLY DELIVERING QUALITY DATA PRIOR TO SLA



Graph depicting hours for schedule completion Before and after automation

## APPRECIATION FROM THE BUSINESS

**“Great find on incorrect metric in the audit report”**