Healthcare Hardware Reliability Reporting System

Katherine Iaquinto
August 2024

About Katherine Iaquinto

Master student in Analytics in interdisciplinary program

- Python for data analysis
- Visualization and front-end development

Data analyst at Hiya with responsibility to maintain and extend data infrastructure for reporting on mobile carrier and customer data

- Cloud data engineering (AWS)

Data engineer at T-Mobile

- Web application for reporting
- Power BI dashboards

Project Goal

A healthcare product manufacturing Quality Engineering organization aims to better monitor the health of hardware modules during manufacturing and assembly.

This organization requested a design and implementation of a new reporting system:

- A scalable, secure, easy to maintain data pipeline that keeps current with automated loading of data.
- An interactive visualization dashboard **enabling the ability to visualize test results as a time series:**

Visualizing the test pass rates and overall trends

Enabling engineers / analysts to monitor specific metric values over time

Dataset

Their dataset is the results of hardware diagnostic testing done during various stages of assembly of products at vendor locations.

These tests have generated a large amount of data going back 5+ years.

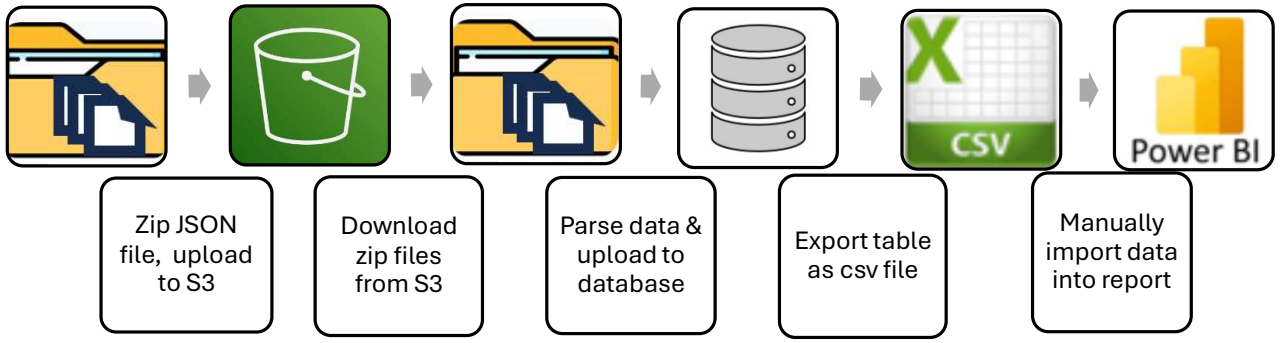
22.7TB Total	100MB Per Zip File	49+ Test Types per Zip

The format and size of the files makes it more challenging to aggregate data over many test sessions, which is necessary to see the larger trends over time.

Previous Data Pipeline

Initially the project scope was to build a revamped version of the last 3 steps of the existing data pipeline diagrammed below.

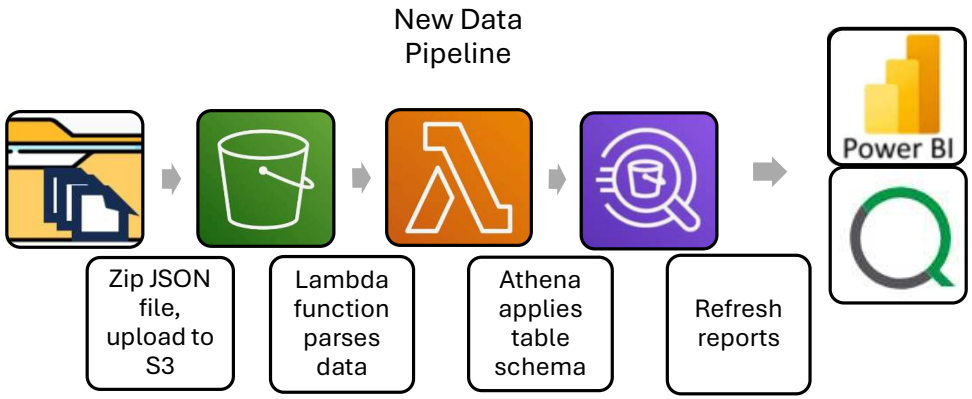
Previous Data Pipeline



- Interviews with the team revealed that the biggest challenge preventing success with the **previous data pipeline was earlier, when downloading the files from AWS.**
  - This procedure was time-consuming.
  - Space on the server was a bottleneck.
  - Data on the server might not be backed up.

New Data Pipeline

After looking at the entire process for ways to reduce waste and increase reliability, the new data pipeline has been redesigned **to reduce the long-term effort expended in upkeep.**



Data processing is now in the cloud (using Python programming and AWS services) to eliminate server space bottleneck. Data keeps itself up-to-date, report refresh time is seconds.

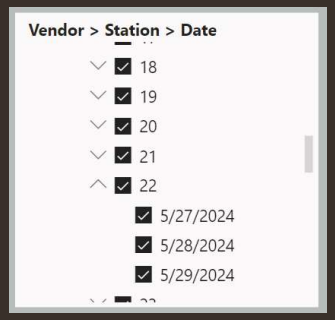
Table schemas were set up on top of the parsed data to enable SQL queries to only get specific data range needed.

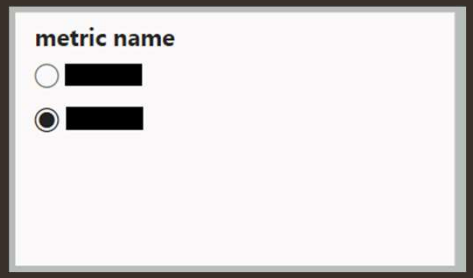
AWS data is set to never expire and additional monthly cost is negligible

Dashboard

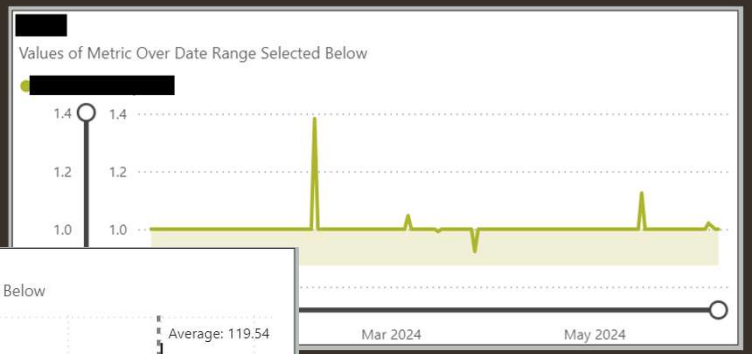
A new interactive visualization dashboard was iteratively designed to showcase insights from the parsed data in Power BI.

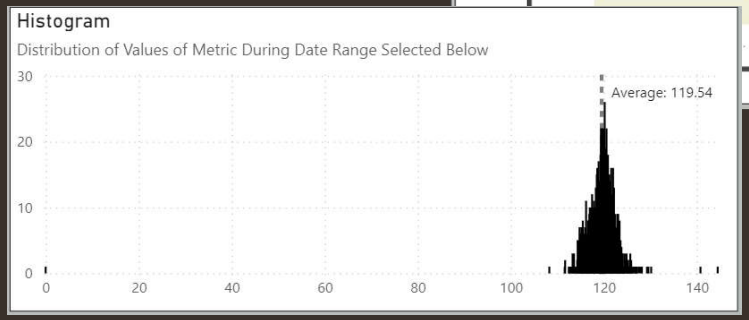
Slicer Options

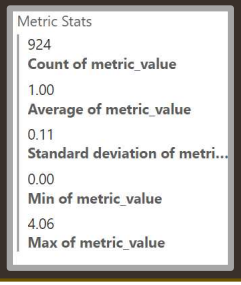




Visualization Options



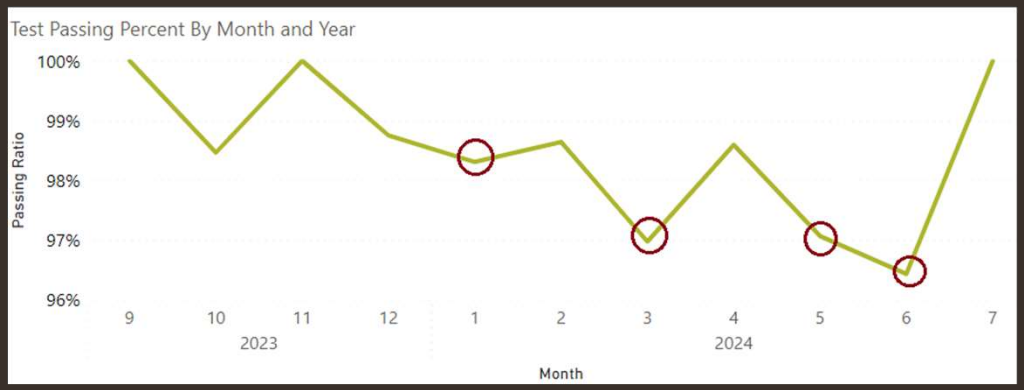




Change Detection Analysis

An additional question from the organization, “How can changes over time in the mean value of measurements be detected?,” led to **experiments in change detection**. These experiments were done with simulated data in Excel.

Exponential Weighted Moving Average (EWMA) Experiment



Results

This revamped reporting system received positive feedback in the client survey and during the presentation to organization leadership.

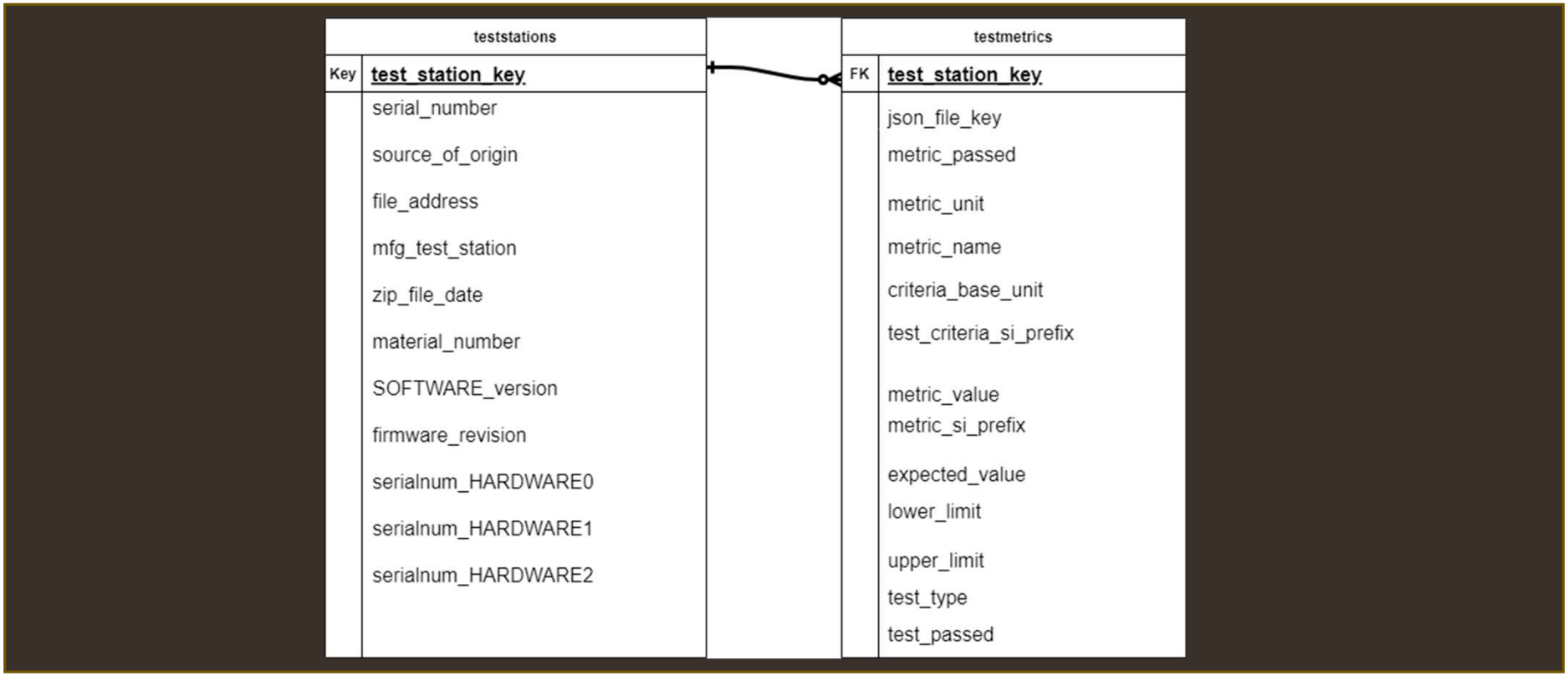
I also received personal satisfaction in crafting a solution to a business problem in an area that is new to me: Learning organization's goals, understanding current state, and designing a solution that targets value creation and elimination of waste

Thank you.

Appendix

Appendix 1

This is a reference diagram for demonstration of the table schemas in AWS Athena.



Appendix 2

Here is a SQL written to impose the schemas on top of the data files.

```
CREATE EXTERNAL TABLE IF NOT EXISTS `ultrasound-test-analytics`.`test_type_metric_unit` (
  `test_type` string,
  `test_passed` boolean,
  `json_file_key` string,
  `test_station_key` string,
  `metric_passed` boolean,
  `metric_name` string,
  `metric_unit` string,
  `criteria_base_unit` string,
  `metric_si_prefix` string,
  `metric_value` float,
  `test_criteria_si_prefix` string,
  `upper_limit` float,
  `lower_limit` float,
  `expected_value` float
)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat' OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://testmetrics-prod/test-type/metric-unit/'
TBLPROPERTIES ('classification' = 'json');
```