The Six Critical Components of Data Testing

DATAGAPS
THE GREAT DATA COMPANY

Data is a precious asset that has to be validated at various stages of use. One stage is at the point of ingestion, another as it moves through your enterprise and lands in your data warehouse or data lake. Finally, when it is consumed in your data analytics platform. This is from the point of view of analyzing data. What about all of the production data that you have in the enterprise? How is that going to be monitored?

So, table stakes for data testing start with access to all the data in your environment, whether in your analytics platform or stored within your production applications. Along with the data access, data quality rules have to be available, as well as a method of comparing data sources of like or mixed data structures and varying volumes, often in the billions.

With these core capabilities, you can develop good testing workflows that take care of 75% of your testing needs. But what about the other 25%? What if your data is in complex hierarchical JSON structures? What if the data testing needs are not anticipated and solved? The last 25% brings about the 6 critical components where you can solve those unexpected needs. So here are the 6 critical components.

Extensibility

In data testing, there are often times when you need to be able to extend your solution to other areas that weren't anticipated. A unique data problem is encountered that is outside the norm and could not be thought of beforehand.

For example, If your solution is extensible through Python or some other method, the issue can be resolved quickly. With Datagaps, we provide a Plugin component that can be selected from a library of components that is extensible by using Python. This eliminates the need for complex workarounds that you have to shoehorn into other solutions.

Advanced API Components

In today's world, data comes to us in a variety of ways. Often as simple as CSV files, feeds from production applications or data that is FTP'd to a location. Quite often, there are requirements to use an Advanced API to get access to the data. In one recent example our client had 8 API's that we needed to invoke to gain access to their Hierarchical JSON data. We needed to create multiple files from each of the API's which meant that we needed advanced capabilities. That is the point of our API component which easily handled the clients needs.

AI based Observability

Writing Data Quality rules is effective in most situations, but often it may not be needed if your solution can learn from the data being ingested. A combination of Data Quality rules and Data Observability is the best approach. Data Quality rules can surface likely data issues efficiently while Data Observability will find outliers that haven't been anticipated before.

Ability to handle large volumes in the Billions

As data volumes continue to grow through a variety of means, at some point in time your normal processing requirements will grow to challenge your data testing capabilities. We provide two means to do data comparisons.

- The first is through a database engine that handles up to 40 million rows and is easier to set up and cost a little less to do the data comparisons.
- The second method that covers high volumes is apache spark based in memory comparisons. This method takes advantage of native cloud capabilities such as clusters and auto scaling.

So if you volumes are low currently the DB Engine will take care of the volumes but as your data scales you have an option to swap out the DB Engine for the Apache spark implementation that can meet your current of future needs.

Integration with your DevOp platform

Your DevOps organization has spent an enormous amount of time and cost to implement a DevOps platform. As you introduce your DataOps platform it is important to be able to integrate with the DevOps platform such as x,y,z. This ensures consistency between how your DevOps ad DataOps process execution and management.

Integration with an RPA Platform

Python, Scala and SQL use cases can be extended to handle a limitless number of variations in your data test plans. However, these languages, while easy to use for developers aren't meant for the business user. Additionally, they aren't designed to mimic human behavior. There is a Billion dollar industry that caters to Robotic Process Automation. In other words, RPA mimics the human interaction.

In conclusion, data testing needs have risen in importance as organizations monetize the use of the data or make critical decisions based on the data flowing through their enterprise. Volumes are increasing, sources take on different access methods, often, data needs to be accessed through alternative means via API or other methods. Your processing needs have certainly grown substantially in the past few years. Methods of testing are changing rapidly. That is why we believe extensibility is so important. As all of these dynamics impact your business and future needs, a platform that will scale and extend capabilities will be critical for current and future needs.

© Datagaps Inc