Why Some Self-Service Analytics Deployments Fail?

Have you ever try to do something thinking you knew how but ended up failing in the end? This is frequently what happens when Self-Service BI is implemented based purely on your past SQL & Reporting experience and is the idea behind this article.

As a principal BI solutions architect, it is part of my job to listen to companies looking to get in to self-service BI as well as to educate them on what I have learned so far from listening to others in terms of what can go wrong!

Self Service Analytics is a way to allow business users & data analysts to answer their own questions by giving them access to various datasets and a very easy tool that can blend, filter & visualize these discrete data sets. If you are planning to roll out a self-service analytics project to a large user base then read the following areas where self-service BI at larger scales will fail compared to using traditional SQL reporting.

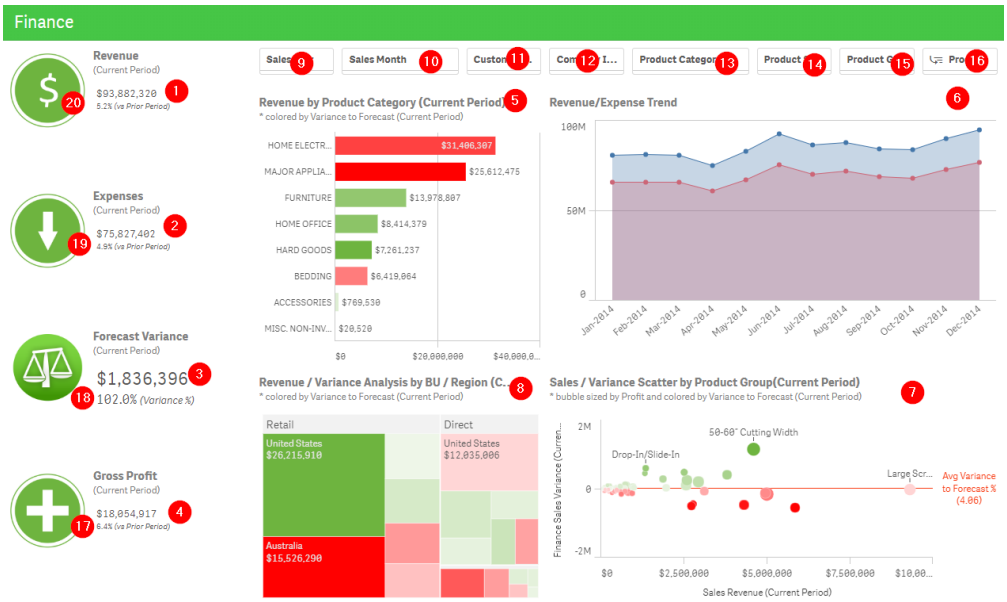
Performance:

When your BI security model is securing access to databases, tables & rows for various users so they can create their own queries, reports & dashboards, you will quickly hit these 2 issues.

Can #1: ***Bad queries = Slow DB Servers.*** If you rely on individual user's SQL skills to join tables & calculate things, then you can expect large number of bad queries. Using incorrect or outer joins, complex calcs, large result sets & more. This will have a major impact on performance of your DB Servers. (SELECT * FROM Orders)

Can #2: ***Interactive Dashboards = Massive Number of Queries.*** Interactive dashboards are cool but can be a major resource hog. Interactive nature of these dashboards means large number of SQL queries being executed with every single user selection and will quickly impact your DB servers when you roll out apps to a large user base. A simple dashboard like below would execute 23 queries with every user selection.

4 simple selections a minute amounts to nearly 100 SQL queries. With 100 users on the system for 60 mins = **600,000** user generated ad-hoc SQL queries per hour. You see what I mean?



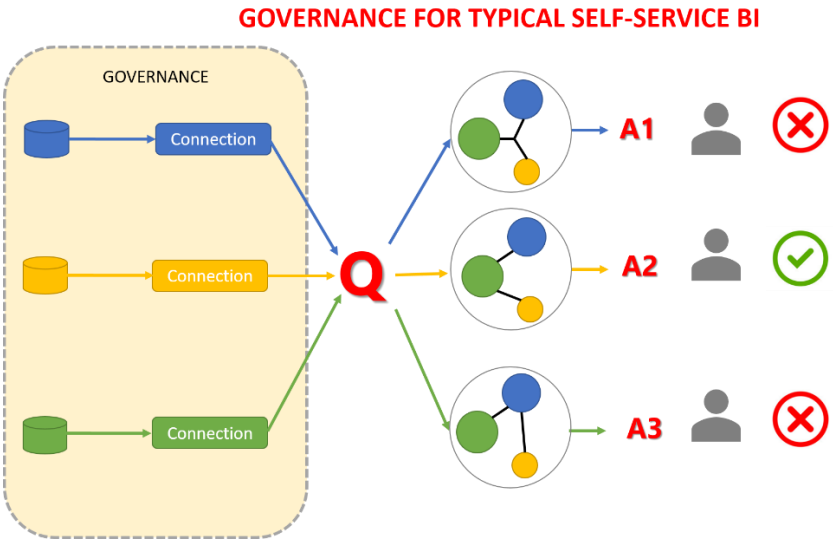
In the end, a data warehouse which had been serving thousands of users using BI/IT managed reports via SQL Queries that had been optimized, tested & verified for years, can be quickly overwhelmed with just a couple hundred of self-Service BI users generating large number of un-optimized ad-hoc queries.

Governance:

You can expect bunch of different results for the exact same thing if you give bunch of users access to bunch of tables from various databases then tell them to create ad-hoc dashboards & reports.

Imagine the number of ways various tables can be joined incorrectly or a same metric can be calculated differently by various users. Business will quickly loose trust in your BI platform If it can't provide consistent results. Inconsistent results will be part of the platform If you can't secure data models and business metrics for self-service. If performance doesn't ruin a large-scale analytics deployment, lack of governance sure will.

Below is an example of how typical BI tools provide governance that secures data sources, tables & sometimes rows. With this approach, same question can lead to multiple wrong answers based on how data is joined & calculated by different users.



Complexity, Security & Scalability:

What will the deployment look like in a year or 3 years in to the future if this thing is wildly successful? Does the solution rely on number of different products and services to address all your needs OR is it a single product platform that does everything you need out of the box? Is it an easy to deploy, horizontally and vertically scalable all-in-one solution or a Lego set that you had to put together? How does it handle multiple authentication methods, larger datasets, more users, Big Data, ML/AI, ETL, embedding, mobile, external facing portals & etc.?

This is where Qlik's Analytics platform comes in. In my next article, I will cover each one of these topics and show you how Qlik addresses them at enterprise scale deployments and show you how it is different compared to other standard Self-Service BI tools. For now, enjoy and learn from other's failures!