Below are proposed steps that companies can take in performing a successful proof-of-concept of QuerySurge – or for performing any software POC.

1) Assemble a PoC team

- You need the right people with the right skills, including data architecture, testing, and moderate SQL skills.

- Keep your team as small as possible, but not too small.

- Identify people who are qualified to lead the implementation and to determine Query*Surge*'s value.

- Not everyone on the team needs to test Query*Surge* in an everyday manner. But it is important to have representation from other groups, such as enterprise architecture and security.

- It is common for a Proof-of-Concept to include managers who think in budget terms.

2) Enlist the QuerySurge team's help

- Choose one person to serve as single point of contact with the Query*Surge* team. A single contact minimizes miscommunication and orchestrates the technical side of the evaluation process. Responsibilities include working with the Query*Surge* team on preparation tasks before the PoC starts, keeping participants on task, and evaluating end results with a final recommendation.

- Schedule a dedicated weekly session where your team and Query*Surge* team work together to configure and execute the PoC scope, scenarios and success criteria.

- Confirm the software budget with your own management. Ensure that the purchase is tentatively approved, assuming the project succeeds.

3) Establish metrics up front

- Clearly document scope and success criteria. One way to go about that is to detail the current pain points – the issues that Query*Surge* is meant to solve – along with how they affect the business. This should include a list of product features, including the must-haves and the nice-to-haves.

- By the end of the project, with the help of the Query*Surge* team, you should be able to provide your management team an ROI calculation to determine how long it will take to reap benefits from the cost of the new software purchase. It is important to have metrics on your current processes, such as resources and/or current time spent testing. For an example of an ROI calculation, view here>>

4) Set a reasonable testing period

- Make sure you create a schedule with deadlines, including a hard stop.

- Dedicate people who have the time (and the skillset) to appropriately evaluate Query*Surge* and let them know its priority.

5) Determine which use case(s) and any add-ons you need to verify

- Use Cases:
    - Data Warehouse testing
    - Big Data testing (Hadoop, No SQL)
    - DevOps for Data/Continuous testing
    - Data Migration testing
    - Business Intelligence Report testing
    - Enterprise Application/ERP testing
- Add-Ons:
    - Test Management connectors
    - Import/Export
    - Ready for Analytics
    - BI Tester
    - DevOps for Data
    - QuerySync

What if you don't have the time, bandwidth or expertise to perform a POC?

QuerySurge Pilot Project
A QuerySurge Pilot Project js a POC program where the Query*Surge* team implements a set of tests based on your requirements and helps you determine whether utilizing Query*Surge* meets your needs for data validation and testing. **more>>**

6) Questions to help with evaluating Query*Surge*

- What are the pain points right now? Can Query*Surge* help solve them?
- Which tasks in the current process does Query*Surge* also need to do?
- Which additional contexts need to be considered, such as security?
- What is a realistic workload to verify?
- What is the impact if Query*Surge* goes down? How do you escalate and mitigate quickly? Which support services are included? Should you procure premium support services?
- Do users like Query*Surge*?
- Does Query*Surge* align with your team's goals?
- What if your company decides not to continue using Query*Surge* after 1 year? What are the options to export your data out of the system?

7) Draw your conclusions

- Work closely with the Query*Surge* team to plan and execute the process.
- Identify the problems you need to solve.
- Try to design the smallest possible amount of work—in a representative environment—to demonstrate whether Query*Surge* would solve that problem.
- Measure the results against key business performance indicators.

If you follow this path, you will have conducted a thorough proof-of-concept and the conclusion should be self-evident and clear.

Thank you,

the QuerySurge team