

## Meeting Your Data Users Halfway Isn't a Recommendation, but a Requirement

*by Neil Martin*

Back when I was still a software developer, I spent a considerable amount of time in the Oil and Gas industry, working for a large blue chip. For one assignment, we built a graphical representation of a refinery as a way to show the movement of oil between tanks. These were separated into two categories: source oils and product oils.

Then, we had to explain to the tank farm operators how to best interpret the visual display they suddenly had access to. "What does this symbol mean? Does it mean the valve is open or closed? Does that mean the flow is going this way or that way?" For them, it represented something of a shift in the way they were used to thinking about their day, so the questions made sense. But we still needed to help them better understand that visual representation.

Data literacy, in the context of an operator on a tank farm, became about understanding what one particular view was showing them at any given moment, in a way that related to their real-world knowledge and expectations.

This was over 30 years ago, by the way. But even if you fast forward to today, I don't think that has changed at all. However, we don't need to live in a world where absolutely everyone is a data scientist in order to properly contextualize this type of visual information. We just need the person working with the tool to be data literate enough to use it properly.

### **To drive data literacy, developers must learn to meet their users halfway.**

From that angle, it's about meeting the user halfway. If we can take the visualization and the product to a point where it's possible to have the user meet us there, and be data literate enough to wield it, then that's what I would call a success. So long as your concept and/or design fits the needs of the users, everything else will fall into place.

### **Unlocking Faster Insight in a More Efficient Way**

Currently, I run a data-driven product development team for a company called [Webhelp UK](#). We're part of The Webhelp Group, which is a truly huge company you may never have heard of. The Webhelp Group is headquartered in Paris, France, and is French owned. We have over 140 locations around the globe, mainly in Europe, servicing over 500 clients in more than 30 languages, across 35 markets. We're one of the top 3 BPO/contact centre outsourcers in Europe, with an annual turnover of 1.3 Bn Euros. Webhelp UK covers the English-speaking part of our customer base. It covers not only the United Kingdom but also India and South Africa. We have about 10,000 people in that part of organization, the vast majority of which are customer service advisors. So, interacting with our clients and customers is what we do all day,

every day—no exceptions.

Our work involves providing tools for use by our internal support and operational clients – one of which is our flagship product, the Performance Compass. We try to help get to a point where a team leader can recognize, for each individual agent, exactly what types of coaching and training they need in order to improve their performance. One of the ways we've done that over the years has involved [Qlik](#), but our process on that front has evolved in a number of important ways.

We've had Qlik in the business for more than ten years now, and for about half of these, we followed an approach that most customers would recognise. We would grab data from lots of different places, funnel it through QlikView, and present it in the form of a document. We then introduced a QlikView Data (QVD) layer as part of our architecture. We then grabbed the data, wrote it to QVDs, and then presented the QVDs in a collection of QlikView output documents. Standard stuff, all things considered.

But we soon found that we had created so many documents, with so many QVDs and at such a rapid pace, that things had almost grown out of control. The QVD layer became so bespoke, and each slightly different QVD so tightly coupled to the output document that was expecting it, that it didn't just make reusing that data layer difficult—in a lot of situations, it made it impossible.

On top of that, we also got to a point where it was tough for new users to understand the information they were being presented with, or in some cases, which document to use. This was putting us farther away from our goal, not closer to it. We had a product, yes, but we saw that we needed to make both the product and the data contained inside far more reusable and more repurposable than our setup allowed at that point.

### **When the Need for Change Evolves Into an Opportunity**

More recently, we've been able to make a number of specific updates to our approach and data infrastructure to emphasize reusability and efficiency. With each operating campaign available in the infrastructure, the data is now in a format that's understandable to everyone. This is true regardless of their current level of data literacy, or even how much they've used our outputs in the past.

One of the recent products that we've been able to build is called "Time Is Money." Both our team, and the Finance Teams we work with, love it. Basically, it converts advisor time or activity time into revenue for the business—hence the rather catchy title. In our contacts industry, there are various possible commercial models, most of which involve tracking advisors' work and how long they spend doing it. That, in essence, is activity-based billing. It's crucial to running a successful business such as ours.

But at the same time, we have to pay the advisors. The same kind of information is used to make sure we pay people for the hours they worked, any adjustments they are entitled to, and that we're accurately tracking absences and other things of that nature.

If you take those two items together—and can measure revenue and cost—you can effectively calculate the financial margin for the operational running of a campaign. We call that "Billed to Paid." The idea was that we could build a single process that would work for any campaign, and with a little configuration we could then deploy it on every campaign in a repeatable fashion. Then, we have a chance to minimize any loss or leakage caused by missing or invalid data using consistent processes.

That's the holy grail, by the way: minimizing leakage. The fact that we're now in a better position to do that than ever is just incredible.

### **Monitor engagement to drive engagement.**

This new approach to our product development has also been massively helpful with user adoption—which is an even bigger problem for us than data literacy adoption. Our latest effort is our usage dashboard. It's a simple dashboard that tells us how users have interacted with the product in the last seven days. We feel that by engaging with some of the users on this level, we can get even more people in the company involved with our data.

#### **Turning Our Attention Toward the Horizon**

In my opinion, the best part is that now that these changes have been made—and have already proven to be successful—my team is in a better position to start focusing on the future. Today, the product is efficient enough that the data can be used for any context and by any client, all while being minutely adapted to their specific needs. Rest assured, that is a tremendously exciting position to be in.

Based on that, we can now begin to iterate on the existing products in a slower—but more methodical—way, while still developing a wide range of products to help create new and more viable revenue streams. We see the next key development as the introduction of Qlik Sense for self-service analytics. We'll use the same data as a source, but we'll choose a group of users who want to add their own spin on things by using pre-defined dimensions, measures, and visualizations. Essentially, we would provide them with a tool they could then use to create their own data story on top of our core data set using a standard document as a starting point.

Cognitive association is also something we're all very excited about. I can see someone in an HR function, for example, looking at some of their personnel data and adding their own visualizations while asking the cognitive engine to describe them. There are some major opportunities there.

Going back to our Performance Compass, if there was a cognitive description that told the user that Person X's handling time was poor last week, but it's a blip caused by Y, that would be incredibly powerful for our users. It would provide better insight with a much more valuable context than what we're currently capable of.

These are just a few of the new opportunities we're currently exploring. I'm very eager to learn about what other ones will reveal themselves in the not too distant future. It's about us, as product designers, figuring out which are the best ones to go after on behalf of our users.

Qlik has also unlocked a new level of adaptability for us, giving us a chance to address new problems in a more holistic way with greater speed and efficiency than ever before.

### **The More Things Change, the More They Stay the Same**

I think it's important to remember that as recently as three years ago, "data literacy" wasn't necessarily a term people recognized. But that didn't make the idea behind that term any less important. When I think back to the time I spent developing software for the oil and gas industry, the tools I used may have been different, and the specific goals that people tried to accomplish varied wildly, but the heart of the matter was still the same.

### **Data literacy = data empowerment. @Qlik**

I was still trying to help people understand their daily lives in the best way possible. I wanted to help them interpret their data, giving them a chance to uncover the story underneath it all. It was about helping people understand what they saw so they could use that insight to work smarter, not harder. That's still our primary objective today.

But the other thing that hasn't changed in three decades is the idea that we must meet our users halfway in between where they currently are, and their ultimate data needs. Yes, we expect more of our users. We expect people to be able to understand the data we give them. But in some cases, that's legitimately too much to ask for—it's too much of a stretch. Meeting them halfway isn't a concession, and it certainly isn't giving up. It's about giving them access to tools needed to help make things easier for them. Nothing worthwhile gets built without the right tools.