



## ***Building the Enterprise Data Warehouse: It Takes a Village***



### **The Problem**

There's no way around it: the good people in analytics, reporting, and data warehousing departments in the healthcare provider industry are getting hit from all sides these days. On the positive side, there is an increased focus on measuring clinical quality with the intent to produce real quality improvement proved by real data. This increased focus is coming from both outside the organization, in the form of additional reporting requirements from the usual suspects (CMS, JCAHO, state organizations), and from inside the organization as clinician-led teams are being empowered and supported to make improvements to clinical quality. All of this is good for the patient, and all of this requires more and more data to measure the outcomes of these efforts. And it's not just raw data that is needed, but complex calculated

metrics that blend information from multiple clinical and administrative systems. So tickets are submitted, request forms are filled out, and/or phone calls are made to already overburdened teams asking for more reports and data extracts.

On the negative side, there is increasing pressure on departments in most healthcare provider organizations to trim budgets or at least stay budget neutral due to shifting patient volumes and falling reimbursement rates. So how do data and analytics departments handle the increased demand (in every sense of the word) with the same or fewer resources? How do we do more with less?

### **The Solution**

First of all, can we do more with less? Yes and no. Let's start with the Yes part because it will be less controversial than the No part.

### ***We're using the wrong tools***

Yes we can do more with less if we use the right tools for the job. For instance, we need to start using reporting tools for reports and stop using them for extracts. You and I both know this is happening at your organization. If a report writer needs to make all the fields on a report one pixel wide to fit all 400 fields on the report so they can be exported to a comma-separated file and submitted to some regulatory or accreditation body, you are not using the right tool for the job. There are ETL tools (like SSIS and Informatica) that are made for this type of task. By using a screwdriver to do a hammer's job you are not getting any of the efficiencies you can get from those tools, and eventually the tool is going to reach its limit and break.

*Some of the things an ETL tool can provide that a reporting tool can't (at least not nearly as well):*

- **Step-by-step processing** – In an ETL tool you can work with your data in an unlimited number of logical steps, whereas with a reporting tool you for the most part need to do everything in one big step
- **Encapsulation** – It is much easier to set up an ETL tool in such a way that if you need to make an adjustment to a single data element, it will only affect that data element and not risk unintentionally (and sometimes unknowingly) breaking other data elements as is more often the case with a reporting tool
- **Exception handling** – ETL tools allow you to handle common issues with your data instead of having the whole extract break every time the common issue pops up again. For instance let's say you need to map your department names to different values required by whomever you are submitting the data to. If a new department without a mapping comes in, with an ETL tool you can set that record aside in a temporary table until someone can perform the mapping.
- **Alerts** – To keep with our department mapping example, with an ETL tool you can set up alerts via email or otherwise to let someone know that mapping needs to occur. Not something you can do with a reporting tool.

Another example of using the wrong tool for the job is when Excel is used as a dashboard tool. Excel is an amazing piece of software that can do a lot of things, but it shouldn't house your corporate dashboards. The main problem with using something like Excel for your dashboards is distribution. There isn't an inherent mechanism for automating the refresh and distribution of the dashboard output to end users, at least without a lot of macros and other crazy antics. So after spending a lot of time and effort wrestling Excel to get it to look and act the way a good dashboard should, a lot of time and effort is then wasted on figuring out how to get the dashboard out to the masses.

A proper dashboard tool (like QlikView, Tableau, and even Business Objects Web Intelligence in certain situations) combines snazzy graphics, user-friendly

functionality, and speed with easy distribution to any number of end users.

#### ***When does the village come in?***

We've talked about how yes, you can do more with less by employing the proper tools for the job. But that will only get you so far. The requests for new reports, analysis, and even whole new datasets are coming too fast and furious to keep up. Gone are the days when bringing billing and administrative data into an EDW and throwing report and ad-hoc query tools on top was enough. Now we need cardiac registries and patient satisfaction and centralized metric repositories brought into the mix. So no, we can't do all of this with less, but we might be able to do it without growing the EDW/BI team or at least without growing it so fast. How? You know the old adage: it takes a village to raise a child—

well it's going to take a village to continue to raise the EDW too.

Think of it this way: what often happens with an EDW is the data architects, BI developers, and system/business analysts come to "own" the data in the EDW. They are seen as the subject matter experts for whatever datasets are in the EDW and they are expected to fully carry the burden of understanding the ins and outs of the data and ensuring data integrity. If "wrong" data ends up on a report, it's the EDW staff that usually gets called to the carpet. But who should really "own" the data? The technologist responsible for making data reportable but whose exposure to the clinical practice that produced the data consists of a tour of the nursing unit that one time? Shouldn't it be the clinical analyst who is meeting with someone in the unit several times a week and

is immersed in the specific dataset in question non-stop?

That the EDW shouldn't own data integrity might still be a novel idea in some places, but with all the new clinical datasets coming into the EDW these days, organizations are coming to realize that the technologists can't possibly maintain a deep understanding of all the data content in the EDW. The data needs to be owned by people who work with that particular subject area every day and have an understanding of the clinical world that data comes from. The EDW team should function largely as the forklift operators in the organization, moving data from one place to another to be owned and easily accessed by people with the subject matter expertise to know what to do with the data and how to tell if it is accurate.

But taking the “village” concept a step further—and here's where it gets more controversial—to be an EDW that keeps up with the ever-quickening drumbeat of data needs and requirements in healthcare, we need to start enabling those outside of the EDW team to actually help *build* the EDW. Not just create reports or even develop more advanced BI tools (semantic layers, dashboards, etc.), but actually help build the data at the most granular level of the data warehouse. Isn't this just making more people data architects and therefore growing the EDW team? No, this is about adjusting the traditional roles of the EDW team and analysts outside of the team in a way that allows the people in each role to do what is most in line with their skills and abilities.

### ***A real-world example***

For example, let's say an organization wants to focus on reducing their preventable readmissions. To do so, a performance improvement specialist is leading a team of hospitalists, care managers, and nurses to evaluate current performance and look for improvement opportunities. Their methodology for identifying an inpatient admission as a readmission comes from a combination of national readmission definitions and their own clinical judgment. Who knows better about how to calculate the field that flags an admission as an official readmission: the EDW data architect or the clinical data analyst with a nursing background who sits on the readmission team? Obviously the latter, so does it make sense for the data analyst to try to translate the logic to the data architect or for the architect to provide the tools for the analyst to create the flag her/himself?

#### **The analyst:**

- Knows how to calculate the readmission flag
- Knows the ins and outs of the data and the “why” of the readmission flag logic
- Has sufficient context to be able to identify right vs. wrong data often just by looking at it
- Keeps abreast of changes to the readmission flag logic and to the workflow inherent in the underlying data systems

#### **The architect:**

- Knows how to “provision” the data, meaning how to pull together the data needed to calculate the readmission flag
- Employs tools in an optimal way to make the readmission flag logic easy to view, update, and keep in sync in all the places it is used
- Can provide both summarized and detailed data to the analyst to monitor the measure and gauge accuracy
- Knows how to tune (optimize) the queries that pull the data to keep the whole process running smoothly and efficiently

The step of the data analyst translating the readmission flag specifications to the data architect is really just waste, not to mention the fact that since the data architect lacks the context surrounding the data, there will likely be extra cycles in getting the flag right that the analyst wouldn't have to do alone. So in an ideal world, we would take out the middleman and allow the data analyst to build the readmission flag. And this would need to happen at the lowest level possible in the EDW so the flag (or insert any other data element or measure here) could be pulled from one place. Attempts to replicate business logic in multiple dashboards or reports might start out ok, but over time and despite best intentions the data elements and measures will get out of sync with each other. The problem needs to get tackled within the EDW at a table and column level.

So practically speaking, how do we outsource some of the development of the EDW to data analysts in our organization? As we've alluded to above, unless you are going to force everyone to use the same BI tool (and good luck with that long-term), the creation of these data elements

and measures will have to be done at the table and column level in the EDW, and this means there won't be a nice user interface for data analysts to use. They will likely need to become comfortable with sql (structured query language—the common language for querying databases). Now once you stop laughing at how far-fetched this idea may seem, hear us out. We have seen analysts who have vowed to never learn sql come around to fully embracing and enjoying the benefits of knowing how to write sql code. Despite all the new tools out there, sql is still the basis for the vast majority of BI and EDW technology.

Here's how it generally works out in the real world (after the data analyst emerges from the Denial stage of course):

1. The data analyst will sit down with the data architect and the data architect will write the sql code for the data element (like the readmission flag) or measure (like readmission rate)
2. The architect will help the analyst become comfortable over time with the sql statement, such as simply running it and viewing results and eventually understanding where the filter criteria can be found (“you can see here it shows the discharge statuses that are excluded”)
3. The analyst will start to make changes to the sql statement and run it past the data architect to validate
4. The analyst will start creating new data elements and measures by using the existing sql statements as a template
5. The data analyst will eventually gain an understanding of sql and will come to appreciate the flexibility and scope of analysis not possible inside a BI tool
6. Peace and harmony will prevail\*

*\*results may vary*

### In closing

Take these things into consideration and see how they might be able to apply in your organization. If you use the right tools for the job, if the right people own data integrity in their areas of expertise, and if you can employ those outside of the EDW/BI team to help develop the EDW, you will be a more efficient and effective organization that is better equipped to handle whatever data demands get thrown at you.