The Need for Project Management in Data Warehouse projects

The Data Warehouse business case

Enterprise Data Warehouses, Hadoop, people swimming in their data lake, the data convenience store / mart (that, in some cases, are undertaken even before you've built the warehouse infrastructure – go figure!) have organizations searching everywhere for their analytical dollar. Data analysis projects for some reason struggle with being able to be rationalized or justified. Yet there is a wealth of data (see what I did there!) to support the justification. The same is also true for PMOs, but that's another story for a different time. But, because of the skepticism regarding the value of data / analytics projects, it is even more vital the projects are sufficiently planned, managed, and well executed.

Value in previous experience

We've all seen the horror stories about failed projects. Missed dates, over budget, under deliver (scope)….. Hmm, the triple constraints rear their ugly head once again. Yet, we continue to see people put into roles who have never managed projects, but because there are qualities about certain roles that may share a similarity to a project manager, we bang home the square peg into the round hole and are confounded as to how this could have possibly happened. Unfortunately, it's the "project manager" and ultimately the project getting banged on the top of the head. It's like taking a dotnet developer and asking him or her to write in COBOL - After all, it's all code, right?

In theory, it is just another project. Well, sort of. I should be able to plug and play my handy dandy project manager into my data project and everything will be fine! In the words of a well-known college football prognosticator, "Not so fast, my friend!" There is absolutely is value in properly managing a data warehouse or similar data-based (not database) project. What is often overlooked, however, is the value (necessity) in a project manager having managed projects in this space before. There are several industries or domains of expertise that, in my experience, warrant a project manager with previous experience in the domain. Large pharmaceutical projects (drug/compound development), where the projects can last upwards of seventeen years for vaccines, need project managers who have been there.

Project duration and cost

Let's be frank. Data analysis projects are typically not inexpensive, or short in duration. There are commonalities with other large projects and programs, but with data warehouse projects, this is the norm, not the exception. The tools tend to be pricey, there is significant demand (read - $$$$) for the top resources. Coupled with a perception of a questionable value proposition, one can see where this type of project would be under increased scrutiny, with jittery project sponsors ready to pull the funding at the first hiccup. With all these influences working against the project, wouldn't you, as the sponsor of such an initiative, feel more comfortable with someone leading the effort who has been there before? Someone who knows what to expect, understands the typical risks (more about that shortly) and is adept at managing them. As with any large, long duration project, setting and managing expectations becomes even more important.

What else is different?

When looked upon as a software development effort, there are many disparities brought to the fore requiring a different approach:

- Corporate Politics
- Requirements
- User experience / interaction
- Code (ETL / Stored Procedures)
- Training considerations
- Risk management

Corporate Politics

Most won't admit it, but politics frequently play a significant role in an organizations information technology strategy. Rightly or wrongly, it can alter what projects get approved and what projects don't. We've all heard of the "pet" projects and we'd like to believe executive decision making would not be driven by emotion, but we'd be wrong. As mentioned earlier; in theory, projects should be approved based on their merit – merit being defined by achievement of corporate objectives, business benefit (incremental revenue or cost out), return on investment, project dependencies, and speed to market.

It is sometimes difficult to quantify the potential business benefit a data management project can provide. In many cases, the driver behind a project and its approval depend upon the quality of the sponsor. It sounds like an odd phrase; the "quality of a sponsor". What does that mean? In essence, what is meant by quality is the effectiveness of the sponsor as a champion for the project. When in the realm of enterprise data management, the sponsor's ability to defend the need, above not only other enterprise initiatives, but also within the data management domain is key to a) getting the project approved, and b) the project completing as planned.

Prioritization occurs at multiple levels; at the project level, but as is frequently the case with data management, within the project as well. With project sponsors, as with any leadership role defined for a project, "There can be only One." There is only one sponsor, one project manager, one person accountable for a deliverable, an issue, or a risk. The project manager has an agreement with the project sponsor, the details of which are documented in the project charter, by the way. The PM is there to protect the interests of the person footing the bill (the sponsor) for the project. It's in his or her best interest to ensure the benefit for the project is well defined and achievable. For that to occur, the PM needs to know "whereof he or she speaks." That is, they need to know what they heck they're talking about.

A PM who has never been involved in a data management project is as an extreme disadvantage and presents a risk to the project (more about risk to come later). Let's focus on the efforts within the data management initiative. Competition for "whose data goes first" can be fierce. One would expect the decision to be merit-based. But again, we're talking about people and when we talk about people, emotion sometimes gets a vote. This is where navigation through the corporate political environment can be trickier than the technical environment. I would almost always argue the former is more formidable than the latter. There needs to be an understanding of the data; its dependencies, availability, quality, etc. Again – this can be problematic for someone without any prior experience in the data management world.

Requirements

At the end of the day, it's all about the data. The use cases and requirements are driven by the data, not by a user's interaction with an application. The focus is typically on "Why do you want the data?" which is driven by "What question or questions are you trying to answer – what is your business problem you believe having this data will help you analyze or solve?" As is frequently the case in this domain, the users typically don't know what they don't know when it comes to the questions that need to be answered. Here's where prior data management experience for both the PM and the BA really shines. For the BA, it's more about prying into the need for specific information, the relationship between criteria and measurement, driving through the data hierarchy in a way that is meaningful to the user – how the data is sliced and diced or how it is aggregated. Do the aggregations make sense? If not, why not?

For the PM, it is, among other things, also about managing expectations and scope. Requirements could quickly spiral and escalate from what should have been a single report to the dreaded "data dump." (Search for "We Need a Data Dump" on YouTube if you're not sure what I'm talking about here). Users are trying to analyze 'something' and solve a problem. Understanding requirements means understanding the business processes and associated data to be analyzed. For a traditional application development project, requirements drive the functionality of the app. If done correctly, the requirements are easily traced to the use cases and the capabilities provided by the system to be developed.

User Experience

In a data management project, the requirements drive the data to be sourced. Yes, you could argue there will also be requirements about the reporting or visualization tool to be used, but you don't normally talk about the "user experience" when dealing with data management projects, in part due to the nature of what is being developed. Typically we are instantiating data, not an application. When your business partner(s) are not sure what they want to analyze or don't completely understand the nature of the data, the "easy" answer from their perspective is, "Just give it all to me, and I'll figure out what to keep and what to get rid of. Can't you just use select *?"

Additionally, how well the problem is defined (in business terms) can determine whether a large warehouse initiative is successful or not. Oftentimes the project team and business stakeholders are not speaking the same language. We're all guilty at some point in time of babbling techno-speak. Anyone who spends a great deal of time with a group of people tends to pick up the habits, mannerisms, and language of the group – even accents! If this communication gap is not bridged quickly, the warehouse can be morphed into siloed data marts which, if developed before or in place of the warehouse, do not lend themselves to enterprise analysis.

Code (ETL / Stored Procedures)

All of this, in turn, impacts the code that is written, which is about pulling the data from source systems, transforming it into a consumable format, and making it available for the users; the Extract, Transform, and Load (ETL). Nowhere in this discussion was there talk about a user interface, location of buttons, screens, etc. Again, the interaction is about how the data will be analyzed. All totaled, there can be significant costs associated with the ETL, storage of the additional data, backups, nightly processing, as well as performance considerations of all the newly acquired data. Remember, above all else, the PM must be a master communicator (channel your inner Ronald Reagan here). It is the PMs primary weapon in his or her toolkit. They must set and frequently re-set expectations to ensure the project stays on track.

Training

There are also training considerations. When push comes to shove on some projects, "crunch time" as it were, late in the project, if something has to give to maintain schedule integrity, training and testing execution are typically at the top of the list. Schedule compression in these areas significantly increases the risk associated with successful completion of the project. But here is where the data project *may* catch a break. Unless there is an introduction to a new analytical or visualization tool, there may not be a need for training.

Risk Management

All projects have risk associated with them. I love it when performing a project assessment, the project manager tells me there aren't any on his/her project, or that they're all being managed, not documented mind you, but “managed”. Whether naiveté (dangerous), ignorance (is not bliss), or laziness (find another PM), previous domain experience lends itself well to understanding the risks facing the project, and data projects are no exception.

Yes, the project manager can be a risk to his or her own project; particularly if that PM has blinders on about their own shortcomings. Quite frankly, most people have blinders on of some sort – it’s called ego. Excellent project managers tend to have some level of self awareness others do not possess. They must be confident, but cautious as well. The PMs goal is effectively mitigate and manage the risks to prevent them from actualizing into issues.

Within the domain of application or system development, not to generalize to greatly, but there is not a significant amount of difference across the domain. There may be differences in development languages, operating systems or database platforms that specific experience may provide more value in managing project risk. The same can be said for enterprise data management projects (in terms of the differences mentioned above). The PM for an EDM project, however, will also benefit from his or her previous experience when it comes to the data. Previous experience with source systems; how normalized a database may be, for example, may impact the effort estimates for tasks. Network constraints may impact where data transformations should take place (ELT vs ETL). Again, previous domain experience builds that internal best practices knowledge base. Obviously, it would be much better if the best practices were documented – risk mitigation if the PM should get run over by the proverbial beer truck.

Project Methodologies / Lifecycle (i.e. Agile vs Waterfall)

Does the methodology or project lifecycle make a difference for data management projects? From a scope perspective, there may be benefits to be gained from using Agile; the potential to provide data for analysis quicker to the users is always more preferable, but the methodology is less important than the domain experience. Obviously for data warehouse implementations, Agile does not make sense since there is much more infrastructure and up front design work to complete. For smaller bodies of work compared to a data warehouse for example; cube development, reporting solutions, visualization, and possibly data marts may lend themselves to more iterative implementation methodologies. Again, this comes back to the prior knowledge of the project manager to understand the scope of work and what the best approach may be to increase speed to market and ensure overall success of the project.

Other considerations

Leadership is the ability to get people to do things you want done because they want to do it; that is, voluntarily. Project managers succeed or fail based on their leadership ability. To have an understanding of the work that needs to be completed, from eliciting requirements, to translating those requirements into actionable data models and design, and moving through the data management development lifecycle, drastically improves the likelihood of a successful data project. Keeping the sponsor engaged keeps the SMEs and other project stakeholders engaged (it's amazing how that works!). Poring over data is not glamorous; it's typically not fun. Structures, models, maps, relationships are all key deliverables that the project manager needs to know why, how, when, and in what sequence they need to be delivered. That is where successful enterprise data management becomes sexy.