Gathering Data Warehouse Requirements

Introduction

Designing a data warehouse begins with knowing the what and the why of the project. Note that using the term "data warehouse" at this juncture is premature as the solution is yet to be defined. This document goes through a requirements-gathering exercise for a hypothetical reporting project. It is fuel for thought, not a recipe.

Define the Mission

Purpose

Work with the users to establish project purpose. Even if you think it's obvious. Get your users to articulate the why of the project. Use their words. Don't correct them. If there is no clear purpose, then the project is doomed.

Users and project management must all judge in context of purpose. The decision for scope inclusion begins with determining a feature's alignment to purpose.

Vision

Crafting a vision is literally an art. The vision is how the users see themselves and the benefits accrued once the solution is in place. Do the users see themselves going home earlier? Or, do they want to impress others with fancy output? Create an image of what success looks like.

Drivers

What's pushing the users to the solution? Does an IT guy just want to build something? Or, has a public company faced an audit finding? Is there a regulatory justification? Decide if it is more important to have a good solution now vs a great solution later. This may be obvious to some but it's important to capture the drivers to make all are on the same page.

Business Case

A project must have a positive business case. Otherwise, it is a waste of company resources. Use a Brief Value Assessment to build a business case. It is crude and informal, but it will get the stakeholders to justify the initiative.

Brief Value Assessment

The following is a brief, rule-of-thumb assessment that can be done in a single meeting. It is completely made-up and lacks MBA substance. Develop the project's value by comparing benefits and costs to doing nothing.

Value = benefits - cost

Benefit Potential

- New business
- Savings
- Opportunities (e.g. ability to scale)
- Risk mitigation
- Regulatory compliance

Costs

- Resources required (people, software licenses, infrastructure)
- Duration to implement
- Ongoing costs (vendor support costs, personnel)

Value

Connect the drivers to the benefits. If the benefits do not align with the business drivers, then the solution is wrong. Next, if the stakeholders agree that the benefits outweigh the costs, then the project can go toward. Note the "margin" between the costs and benefits. If it's narrow, then the lowest cost, lowest risk solution is in order.

Reporting Requirements

The following is a simple framework for building up reporting requirements. The framework is meant to be practical rather than exhaustive. It should get you and your users talking. It is based solely on experience. There are no fancy Excel deliverables at this point. That comes later. Besides, if you, the beloved reader, are a consultant, then I am sure you already burdened with fancy templates.

What do the stakeholders need to report?

This can be harder to answer than what it seems. Many times, users think they know what they want. But, when pressed, users (aka people) are unable to articulate specifics. Be patient. Refer to the vision if necessary. That's what it's for. It may be that what they want is nowhere in the organization. So a small reporting initiative balloons into a whole new system implementation.

1) Why do they want this report?
    a. How is their future-self better after having this information. May seem deep for a little report, but go with it. The question will scale up or down as needed.
    b. How are they vulnerable today without having the information. Make sure the two responses tie out. Often, the thing being asked for will not actually help with the essential pain.
    c. Ask more probing questions. If the business case is still in flux, then this could help bolster it.
    d. Make sure there's not more to it. If the reporting requirement is in response to a business process failure, then direct the conversation to improving the process. No use putting lipstick on a pig.

2) What are the key metrics?
    a. What are the important measurements?
    b. Why are they important?
    c. Why have they not been reported before?

3) What is the essential attribution or entities?
    a. What is the context for the measurements?
    b. What is the level of detail that needs to be reported? (e.g. department, or by day)
    c. What are the time-comparison requirements (e.g. today compared to this time last year, month-to-day, year-to-date, etc.)

4) What are the special summaries or groupings
    a. Will the report be aggregated in some way?
    b. Do the users still need access to the underlying data?

Who are the people involved?

Data ownership and custody are key data governance issues. Answers to these questions will establish the foundation for all requirements going forward. These are generally a limited concern for small projects, but can quickly balloon in medium to large sized initiatives.

1) Who owns the data?
    a. What department or entity produces the data?
    b. Are there license fees for external data?
    c. What are the data quality guarantees?

2) On whose system is the data stored?
    a. Is the data collection system owned by a different group from who creates or obtains the data?

3) Who will consume the data?
    a. Is the data meant for a different set of individuals

What are the data quality requirements?

Knowing what the data is for informs this conversation. If the intent of the report stops with the immediate team or department, then don't go crazy on data quality. If there's a chance down the road that the data will be combined with some other system, seriously consider dealing with quality and consistency at that time, so long as the data is good enough for today's needs. Building beyond the need *might* help later, but it *will* erode the business case today.

1) What is the expectation for the metric quality?
    a. Is the data acquired *after* an approval, or
    b. Is the data acquired *before* an approval so that the data may be reviewed?

   c. Is the data expected to change after it has been reported?

2) What is the expectation for the attribution quality?
   a. Are the attributes expected to be in terms of the source system, or
   b. Are the attributes expected to conform to an organizational standard (e.g. a single customer list)?
   c. Is the source data of poor quality and require clean-up?

How will the solution acquire the data?

This topic clarifies how the information will be obtained. Data acquisition informs overall solution complexity. If data can be left in place then either built-in application reporting or a report server will work. If the data is expected to be moved for snapshotting purposes then the solution becomes some outboard storage thing.

1) Is data to be reported directly from the source system and left in-place?
   a. This may be too solution-oriented, but confirm if the question can be answered early
   b. What is the source system's data standards (e.g. database, data model documentation, licensing restrictions on accessing data directly, web service API, XML interchange format, etc.)

2) Is the data expected to be moved to an external database?
   a. Why does the data need to be moved?
   b. What are the timings for the data movement?
   c. What is the limit on how long the movement can take?
   d. How much data is moved?

How will the users consume the output?

The following is a list of a-b questions. The intent is to go through the questions with the users and begin to rough-in the solution in terms of the users' vision. Be careful not to slip into technical details. The questions are meant to be answered in the ideal case without all the technical stuff.

1) Electronic or paper (or plastic)
   a. Electronic – leaves open possibilities of web delivery, PDF, mobile, or all of the above
   b. Paper – presentation-worthy, high-quality both in terms of look and data correctness

2) Static or interactive

   a. Static – one-size-fits-all presentation (an extreme, but maybe fits)
   b. Interactive – sort, filter, group, and aggregate at will (leads to questions about underlying detail and delivery tools like Excel)

3) On demand or scheduled
   a. On demand – the report / web page is generated in real-time. Can place a burden on source systems and may lead to data quality concerns
   b. Scheduled – data is either acquired on a schedule or the report is actually generated on a schedule.

4) Structured or ad hoc
   a. Structured – data is produced from saved queries
   b. Ad hoc – users are looking for a tool to issue flexible (possibly advanced) queries

5) Fancy UX or Data only
   a. Fancy UX - Are the users actually the general public or C-suite, if so, then the UX is paramount
   b. Data only – Users are themselves analysts and just want the data and for you to get out of their way

6) Structured or unstructured?
   a. Structured – data is in a tabular, relational, XML, or some other regular format
   b. Unstructured – data is in the form of documents, images, or some other irregular format

7) Final stop or processed
   a. Final stop – the sole purpose of the reporting solution is to shed light on data and that's it
   b. Processed – the reporting solution may actually be the first step in an overall analytics requirement. Purpose and mission is key. If the answer is a surprise, this could lead to important scope discussions.

8) Summary or detail
   a. Summary – is the data summarized at some level and nothing more?

   b. Detail – if there is any hint of drill-down or flexible UX, then maximum detail is needed. Sizing, throughput, and response times are a concern.

9) Desktop or mobile
   a. Desktop – data is delivered electronically and restricted to the corporate network
   b. Mobile – could mean just a mobile-conformant web page, or an actual mobile app. Note that shipping data outside the corporate network entails all sorts of security and networking concerns.

Detailed Mockups

Mockup the end product using Excel or PowerPoint. Again, do not get bogged down in the details. If there is data movement involved, map out a process flow that shows the coordination between business and system activity.

Timing Needs

Timing requirements can be deceptively complicated. Think about the flexibility of the recurring appointment feature in Outlook.

1) When will the users expect to see the report?
2) When it’s the data available?
3) Is the report “live”, i.e. operates on active transactions?
4) Does the data need to appear in an inbox?
5) Is the report event-driven?

Downstream Systems

What other departments or organizations expect to use the data?

Are they firm requirements or a nice-to-have?

If they are hard requirements, are there “single-source-of-truth” concerns (e.g. multiple customer lists or varied standards)?

Non-functional requirements

The following lists all the essential system requirements and conformance standards.

1) Throughput Performance
    a. How soon do the users expect the data to be available?
    b. How much data is there?
    c. What is the technology's capability? How fast can your server save a 1GB file to disk? To a network drive?

2) Response Time Performance
    a. If the solution is interactive, then the expectation is instantaneous screen refreshes, no sense in asking users about what an acceptable response times is.
    b. Mockups are important for informing how much data is involved for displaying results
    c. Therefore, response time should be stated as *system must present x data in 1 second*
    d. How many simultaneous users must the system support?

3) Sizing, Storage and Retention
    a. If the data is outboarded, how much is there?
    b. For how long must the data remain available?
    c. Can data be archived to cold storage?
    d. What is the acceptable amount of lead time between submitting an archive request and receiving the data?

4) Security
    a. Access to the Reports – who will need access to the reports
    b. Row level security
    c. Data at rest protection
    d. Data in motion protection

5) Confidentiality
    a. Personally identifying information
    b. Sensitive data
    c. Obfuscation

6) Availability, Business Continuity and Remote Access
    a. What are the organization's business continuity requirements?
    b. How quickly must the system be up and running in the event of a disaster?
    c. What do users need in terms of remote access?

7) Regulatory concerns
    a. Is the solution needed for regulatory purposes?
    b. Must the solution conform to regulatory requirements such as SOX or HPAA?
    c. Are there internal / external audit requirements?
8) Existing tech stack and skills
    a. What technology and tools already exist in the organization?
    b. What is reaching end-of-life and needs to be upgraded?

Conclusion

Requirements gathering can be fun. At this point in the project, everything smells of lavender. Work hard to listen. Repeat back to your user what you have heard. Avoid getting mired in technical details. Get sign off on the requirements but expect things to change as the project moves along.