DOQS Data Warehouse Primer

Richard E. Biehl, Ph.D.
Data-Oriented Quality Solutions

General Concepts

A general multidimensional data warehouse contains two types of tables: fact tables and dimension tables. A fact table is a very large centralized table that contains specific measurable data about the business. The fact tables are defined, and surrounded, by a collection of dimension tables that represent the natural dimensions within which the facts are defined. The dimensions are said to provide the context for the information in the central fact tables.

By contrast, a relational data warehouse represents the historically traditional data base implementation of the data warehouse concept. While the warehouse contains many tables, no explicit separation is made between fact tables and dimensional tables. Business information is contained throughout the warehouse. Because the data is distributed, more knowledge of the data is needed in order to build queries, and the database can only be optimized for certain families of queries. Queries that access data through less optimized paths tend to cost more, run longer, and can be much more difficult to design.

The dimensional data warehouse offers significant improvements and benefits over the older relational approach. To add new data to a relational data warehouse, it typically requires designing new tables and columns in the database that will store the new data. In the worst case scenario, new data might be in conflict with the warehouse's existing design, necessitating a complete redesign, or a foregoing of the new data altogether. This usually requires a full information technology project to design the changes and coordinate the implementation. By contrast, dimensional data warehouses are remarkably stable over time. Typically, most new data can be added to a dimensional warehouse without any database changes at all. It still typically requires an IT resource to implement new data, but the effort required, and impacts endured, are typically an order of magnitude smaller than the same change in a relational environment.

The ease with which information can be added to the dimensional warehouse is what provides for the excellent flexibility of this approach, and enables a migration path for users that avoids the major expense and effort of a "big bang" implementation. The warehouse can start small, with only a few critical data feeds; with new data being added in successive releases with minimal impact on early users. This also supports changes and shifts in the actual source application systems that provide data to the warehouse; whether shifting from diverse uncoordinated systems toward a more centralized enterprise solution, or through the addition of specialized niche applications that are needed to round out the data in the warehouse. Release schedules are typically measured in months, where similar changes in a traditional relational warehouse can take years.

Unbeknownst to most organizations, an in-house multidimensional data warehouse has existed for years: the general ledger. Planners and users of the general ledger have vast experience in dealing with multidimensional data definition and aggregation, control of dimensional reference data, and integration of multiple and diverse data sources. In the case of a general ledger, the variety of dimensions is typically hidden in the account coding structure of the chart of accounts; but other than that small difference, the general ledger is a big fact table in which each individual fact is provided context by the internal (dimensional) structure of the account number. Managerial accountants used to navigating and aggregating the diverse data that ends up in the G/L can play a significant role in the deployment of data warehousing technology in an organization: The fact tables are the G/L, the source data feeds are the subsidiary ledgers, and the chart of accounts provides the dimensions.

A user of the general ledger is already prepared to face many of the challenges confronting new data warehouse users. These challenges only appear new as data in the warehouse expands beyond the accounting arena. Viewed against the planning and management requirements of managerial accounting, many data warehouse design issues begin to look more conventional, if not routine; and the generalized dimensional warehouse model allows the focus to remain on the decision support requirements, not the information technologies needed to meet those requirements.

Because of the stability of the dimensional warehouse model, the costs incurred to implement new releases of the warehouse over time, usually with each release adding additional data feeds to the warehouse, are much lower than might be anticipated by comparing such release development to more traditional data warehousing or information systems projects. Once the startup costs of the first release have been incurred, two major elements of future costs have been eliminated.

First, the actual database structures that support the warehouse are extremely generic and stable. It is extremely unusual to require extensive database changes as new releases are introduced over time. At Mount Sinai Medical Center in New York, the data warehouse built on this model has not required any redesign activity since the first release was introduced with seven data feeds three years ago. Today that warehouse receives over 50 feeds, and loads them into a database structure that has not changed since the initial release. Second, the ETL architecture needed to load data from multiple sources into the warehouse also stabilizes during the first release.

As a result of this initial investment in database design and ETL architecture, subsequent warehouse releases require far less marginal investment than might be seen in other warehouse or system architectures. Relational warehouses, in particular, typically require additional tables and indexes for each new data feed implementation; resulting in new database design and ETL architecture investments being required for each subsequent warehouse release.

Data Warehouse Integration

The core approach to storing and organizing the financial and non-financial information needed to support effective decision support is the overall integrated data warehouse, a multidimensional star-schema data warehouse implemented on a mainframe or network server platform using one of the large-scale commercially available relational database environments. The data solution provides extensive flexibility through the use of financially, clinically, and organizationally specific analytical dimensions that can be used to drill-down, focus, and tailor individual decision support queries to a variety of organizational levels of detail combined with a variety of clinical and operational factors.

The flexibility of the design enables the data warehouse to be configured to handle a variety of levels of detail and granularity without significant change. The range of queries possible against the data is most directly influence by the levels of integration chosen for each of the major data sources provided to the ETL environment. Over the long-term, the model treats financial data as the backbone of the data integration strategy, and so at a minimum, the model expects to receive financial data for the organizations and facilities within the configuration scope; including data analogous to general ledger accounting and subsidiary ledgers for plans, budgets, and actual results. This data alone provides a great deal of decision power when modeled into the warehouse, as evidenced by most organization’s emphasis to during early implementations on financial warehousing. Managerial accountants will recognize that the dimensionality of the warehouse provides a way to slice-and-dice loaded data; in ways analogous to the slice-and-dice made possible by the multi-segmented account numbers in the organization’s chart of accounts. The extra clinical and operational dimensions of the warehouse simply add additional depth to that chart of account structure depending upon what other data is provided through the ETL feeds.

Having expressed requirements related to the integration of clinical data, the organization will need to obtain clinical data at some level of granularity and integration to the warehouse for loading. At the lowest level of integration, the warehouse can be loaded with count and aggregate information from the clinical systems; enabling financial queries with very simple denominators (i.e., per patient, per encounter, per visit, per provider). As more data is provided, the level of integration is raised, and more sophisticated queries become possible (i.e., per male patient age 40-55 with both cardiovascular and diabetes diagnoses, per female patient with at least one live birth in previous 3 years, per provider by clinical specialty and years of experience).   As the data integration is increased, the ability to conduct analyses that look at finer and finer nuances in the data become possible, enabling the identification of financial and operational patterns that likely affect only subsets of the organizational, provider, or patient population.

It is this ability to slice and segment data across the warehouse that gives the integrated dimensional data warehouse its power. As a dimensional warehouse, queries that can be imagined against any dimension become readily usable against any of the other dimensions. A query designed to measure productivity or profitability of a physician (in the Caregiver dimension) can just as easily be adapted to measure the utilization and profitability of an endoscopy scope (in the Material dimension), or a surgical theater (in the Facility dimension). At the most extreme level of integration, all of an organization's clinical detailed data can be loaded into the warehouse, maximizing the analytical sensitivity available to users, allowing for analysis of financial and other factors related to disease management, population health, physician and nurse staffing, facility management, patient safety, length of stay, or case management.

Depending upon longer-term requirements, an organization might eventually want to move beyond the integration of clinical and financial data for analysis; expanding the range of data to include logistic, operations, or research data; each with the variety of integration levels discussed above for clinical data. Possibly as part of a comprehensive ERP roll-out across an organization, this broader range of data would enable analyses that go beyond patient care to include purchasing and inventory facts, engineering and housekeeping impacts, as well as clinical trials and grant funding for studies. Analyses are limited only by the data made available for loading.

The power and advantage of the integrated dimensional data warehouse design is that it does not require any particular up-front commitment to a particular range of data, or to a level of integration for any of that data. The warehouse design is made flexible so that it can store any and all of these types of data, and any level of integration, without design changes. This provides an organization with a migration path that can start small (i.e. financial data with low-level clinical data integration) and continually grow into the future. The organization might choose to add additional breadth and depth to the data over a release migration plan that could take many years to implement, all the while enjoying the benefits of analysis available using whatever data has already been loaded. One medical center implemented this model into production with only 7 data feeds and about 30 users; growing to 45 data feeds and 200 users over a gradual 3-year roll-out; with the system in productive use throughout each upgrade. The design of the warehouse never changed; the information in the warehouse simply continued to grow.

Generic Concept of Operation

This abbreviated Concept of Operation provides an overview of the interaction of the major components of a general data warehouse solution. The general approach to data warehousing adopted by any particular organization can be expected to be similar to this generalized approach.

The integrated warehouse solution is based on a generic architecture that has been proven in multiple settings, and that can be implemented and configured at any time while the source applications that will feed data to the warehouse are being design or implemented. The component architecture is based on industry best practices for data warehousing and business intelligence, and are not varied for individual functional settings. The warehouse architecture is the optimal solution for any domain, eliminating the risk that the warehouse might not be able to be implemented because of domain-specific or application-specific issues or concerns.

The core of the solution architecture is a dimensional star schema data warehouse that has been matured through implementations in multiple financial, logistical, clinical, and operational settings. The warehouse is a standardized generic design based originally on the work of Ralph Kimball, and then extended to include functions and features that support master data management integration as well as the incorporation of semantic ontologies that extend the data beyond that which is available only in the direct data sources. The generic nature of the design allows for implementation in any of the industry standard database environments (e.g., Oracle, DB2), and offers fast and early implementation.

Typically the implementation includes one or more data marts that are created and maintained from the core data warehouse. These data marts are intended to serve a variety of expected requirements, such as unique organization or aggregation needs, altered units of measure, or different performance expectations. The design of any single data mart will be logical or physical, with logical data marts still existing within the data warehouse but accessible as a distinct view, and physical data marts existing as stand-alone database structures. The distinction between logical and physical data marts is mostly transparent to users, and can be done in any of the major database environments; even a different environment from the primary warehouse (although this is rare).

In some cases, the data marts are made *appendable*, meaning that users want to be able to add their own data to that provided in the data mart, and then load that new data back into the warehouse for wider availability and use. Such requirements often appear in data related to planning, budgeting, or auditing where notations made against existing data during the process are wanted in the permanent warehouse for future analysis. Research projects often want to add their own data and notes to a warehouse based on experimentation using data in data marts created from the warehouse.

The data warehouse and data mart environment enables business intelligence interfaces that are the user-centered core of the system; the front end. While the architected solution is agnostic with respect to any particular technology that must be used, one finds that most users prefer a web-based query and report interface that can be placed directly into the web-based environment in which they use the various application systems that provide data to the warehouse. The architecture also offers other client-server alternatives that help meet requirements for complicated non-standard queries and reports, as well as any need for faster performance for those non-standard uses. The most complicated uses might even require externalizing an independent data mart specifically for use in external analysis and reporting tools (e.g. SAS, SPSS).

The back end workhorse of the solution is the Extract, Transform, and Load (ETL) subsystem that brings data into the warehouse from the various user application sources available. While there is typically something distinct or unique about every inbound source feed, the out-of-the-box design patterns for ETL allow for quick configuration of new data feeds into the warehouse. Rather than every feed being considered a distinct flow from source to warehouse, every feed goes through a series of staging areas with each one being more generic than the previous. By mapping data generically as it is processed, multiple sources can converge into a few distinct loads into the warehouse. This allows future data sources to be implemented very quickly because they need only be designed through to the generic stage into which they map. In essence, the closer the data gets to the warehouse, the more it has been transformed into the generic architecture that is the core of the warehouse design. This high level of reuse allows for faster implementation and earlier availability of each new data source.

In support of the mainline flow of data through the ETL subsystem into the data warehouse for presentation in the business intelligence toolset, the architecture invests heavily in the definition and control of the reference data needed to integrate all of the data from multiple sources together; referred to as master data management (MDM). MDM is a weak link in the design of many data warehouse solutions, typically because its complexity is underestimated, or the risk of doing MDM poorly are not understood. This solution recognizes the critical role played by master data in aligning the various dimensions of the warehouse so that data from disparate sources can be integrated seamlessly. Most MDM controls are available as out-of-the-box features that require no extra implementation time, and that ensure high levels of integration across multiple data sources.

At the enterprise level, MDM includes the organizational structures and financial charts-of-accounts used within individual financial systems, as well as built-in data lifecycle capabilities so that forecasted, budgeted, incurred, reported, and adjusted data can be integrated even if sourced from different application systems. All master data is identified by surrogate keys, allowing data elements from different applications to be kept separate in the warehouse even if they were inadvertently identified by the same key values in their original systems. It also supports the same data identified by different keys in the source systems being stored as a single data stream in the warehouse. If two data streams that should have been together are inadvertently stored in the warehouse as different streams, the merging of such data is handled, once discovered, as an automated control out-of-the-box.

The MDM controls also include features to prevent data from multiple sources from colliding in unexpected ways, including controls for unit of measure alignments (including multiple currencies), as well as multiple codeset tables from different systems having slightly different overlapping and non-overlapping values. Lastly, the MDM controls include an ability to incorporate any number of available semantic ontologies, such as the clinical Disease Ontology or Phenotype Ontology, as well as the financial ISO/IEC 15944-4 standard accounting and economic ontology based on the Resource-Event-Agent (REA) Enterprise Ontology. By including such ontologies within the master data of the data warehouse, this solution offers users an ability to query and report data from a large variety of perspectives that were not even envisioned in the data design of the applications from which the data was collected.

Recognizing that even the newest and best application systems still encounter data problems, the warehouse solution includes a component for managing data quality, the intent of which is to keep all loaded data available for querying while also making any quality problems associated with some of that data visible. Quality controls include continuous change control management that allows all of the data in the warehouse to be stored against the reference data associated with it at the time of creation, even though much of that reference data might have subsequently changed.

All data arriving in the data warehouse is also edited and validated against numerous rules, many of which are available as out-of-the-box processes. These checks include whether individual data fields are required or optional, data type and unit-of-measure validation, translation of values to consolidated cross-system values, as well as valid ranges of values. Additional source-specific rules are implemented as needed during the configuration of the warehouse. Failure of any data validation results in the decrementing of a *quality score* associated with the value, but the data is continuously available to users.

One of the most important quality controls in the solution is the dimensional orphan processing. An orphan is a reference value that has arrived at the warehouse from a source, but the validity of that value has not been confirmed through the normal channel that would define such reference values. An example might be an invoice being loaded that indicates a vendor not yet received through the data feed that would load new vendors. While many warehouse applications would reject such data completely (sometimes loading the transaction into suspense), this solution immediately loads the data while defining a surrogate entry for the missing reference data. When the missing data finally arrives, the orphan is converted in a process known as *auto-adoption*, and the decremented quality scores for the related data values are adjusted. Since most orphans eventually auto-adopt, this feature allows for interdependent data loads to be scheduled dynamically without needing to adjust or reschedule loads because of data dependencies. Data availability is maximized, while the need to reprocess suspense data is virtually eliminated.

The final set of controls in the data quality component involves identifying and annotating potential problems in the data at the time series level, even if the distinct values pass all validity checks. Time series analyses include checking to see that required events have occurred (e.g. there’s an order for every invoice, or an order for every result), and seeking statistical outliers that might be of concern to users. Outlier controls are very good at spotting data changes in one data source application that have not been correctly implemented in another, such as a change in ordering units of measure on the procurement side without a corresponding change in the billing units of measure on the consumption side (e.g. booked charges suddenly change dramatically, usually seen as a 3-sigma outlier in a control chart created from the warehouse).

Beyond the control of data quality, the warehouse includes numerous other internal controls that help protect and control the use of the data by authorized users. Access security for the warehouse is typically accomplished through an organization's standard security environment, although custom access control features can be designed as required. Beyond access control, our solution also includes capabilities to control access to data through consent management. Regulatory consent controls include HIPAA criteria management. These controls are built-in to the solution as out-of-the-box features. Non-regulatory controls, such as who can access financial data for different organizational or profit center units, are accomplished through the same mechanisms, requiring some customization based on unique requirements for control.

The integrated ETL, data warehouse, data mart, business intelligence, MDM, data quality, and internal control components provide a complete solution that can be implemented independently of, or even earlier than, the implementation of applications systems that will serve as data sources. Rather than avoiding the complexities that can make data warehousing difficult in such large-scale and dynamic organizations, this generic design *embraces* such complexity by anticipating and allowing for the various data conditions that typically prove problematic (if not fatal) to more traditional warehousing solutions. Once implemented, with only a few data sources, the addition of each additional data sources gets easier, faster, and less expensive. The generic nature of the warehouse design means that queries written against early versions of the warehouse with limited data loaded will automatically expand their coverage to include more data as it becomes available in the warehouse. The notion of the *big bang* implementation, or the need to have all desired data available and loaded at the time of a single implementation, is not necessary with this design. This design brings components on-line early and opportunistically; maximizing data availability and value to users.

Typical Recommendations

The following are generic recommendations that often apply to a new data warehousing client organization. Many healthcare information technology organizations lack the work teams and staff competencies implied by these recommendations. In cases where such groups and competencies already exist, data warehouse implementation can be accelerated beyond the initial schedule outlined below. These recommendations are presented here for primarily discussion and background purposes.

Data administration

Most client organizations have a need to look more carefully at how they establish, manage, and monitor data standards and issues across all of information technology. A new warehousing environment will expand this need, and in order for data administration to get the attention and focus it needs in the future, a more formal charter and structure is typically needed.

Recommendation #1: Establish a centralizing data administration function within the overall Information Technology organization to define, and advocate for, standards and practices related to data and metadata management across the enterprise.

Creating a data administration function is initially about formalizing standards and practices that are at least partially in place, if only informally. The participants should be almost exclusively drawn from the staff of information technology, and their emphasis should be on setting data standards and controls necessary to enable the policies to be set by the new data governance group, and implemented as easily as possible.

The basic responsibilities of a data administration function might include:

- Acting as a communication conduit to and from constituencies to the data governance group on data management issues.
- Representing constituent groups in the broader organizational community on data related matters.
- Promoting best practices for the management and use of enterprise data.
- Communicating with constituent projects and reporting progress and implications of new or updated data policies and procedures.
- Acting as project advocates and change agents.
- Expediting and facilitating data management issues resolution.
- Chairing and/or participating on working subgroups as appropriate.

Participating in meetings or as designees to project reviews. There is a wealth of information available in the literature and industry to assist with implementing this recommendation. The Data Administration Management Association (DAMA) is a good place to start.

Data governance

There are typically no central governing bodies across new client organizations to manage the policies needed for data ownership and definition across the array of applications that are likely to interface with the new data warehousing portfolio. Without some form of governance body, information technology staff will be forced to make choices among data alternatives that should really be made within the clinical user community.

Recommendation #2: Establish a cross-functional data governance function across the organization to set policy and guidance for data definition, ownership, and stewardship of the strategic data asset of the enterprise.

Implementing this recommendation will be much more difficult than the implementation of a data administration function precisely because the stakeholders of interest are almost exclusively outside of information technology. This group of people should be the system and data owners for all of the major data application and subject areas. Their mission is to define data policy, and this often involves arguing among themselves on key issues of conflict. The advantage of such a group is that it prevents information technology from having to decide issues that might stay in conflict if not agreed to by the user community.

The goals of the data governance group might include:

- Enhance the quality of data throughout the enterprise environment.
- Identify and promote innovative uses for data and information.
- Provide advice and information to data owners, stewards, and users.
- Ensure timely and appropriate addressing of data management issues.
- Promote the voice of the customer in data administration products and services.

The objectives of the group in meeting these goals might include:

- Develop data management policy recommendations and procedures.
- Define roles and responsibilities for data across the enterprise.
- Identify enterprise, regulatory, and statutory data elements and controls.
- Promote creation and use of standards and principles to support data integration.
- Establish guidelines for data modification, migration, and archiving.
- Promote acquiring and sharing of data to minimize cost and maximize reusability.
- Develop a vision and recommendations for anticipated future data requirements.
- Develop and share consistent technical solutions for priority issues.
- Conduct data quality assurance analyses and audits.
- Respond to additional data management issues as they arise.

The data governance group will play a key role in prioritizing the data sources available to the data warehouse, working to resolve semantic conflicts among those data sources, and (particularly important in the beginning) reconciling the host of reference data codes that will be in conflict during the initial loading of most of the dimensions in the new warehouse. The success of meeting this recommendation is a critical success factor in successfully implementing the new warehouse portfolio. Without effective governance, the new warehouse becomes a large repository of data that can't effectively be integrated into useful information that can be shared and reused across the enterprise.

Proof-of-concept

In order to make final choices regarding the enterprise architecture for data warehousing at a new client organization, it is helpful to actually build a small-scale data warehouse along the lines of the anticipated architecture to serve both as a demonstration project and as a staff learning tool. It will take considerable time to put in place the necessary data administration and governance functions needed to make key decisions regarding a first production-level warehouse implementation. In the meantime, a proof-of-concept will allow for some experimentation among alternatives, and some visibility for socializing the design intent.

Recommendation #3: Define and develop a prototype clinical data warehouse, of limited scope, in order to develop and clarify needed information technology practices and procedures for data warehousing in conjunction with the implementation of nascent data administration and governance capacities.

This recommendation should not be interpreted as trying to implement a "quick and dirty" data warehouse. The purpose of this prototype is to explore and experiment with the many and varied design alternatives still available with our standardized generic warehouse architecture. For most data sources, there will be multiple alternative ways in which data can be mapped into the warehouse. In this prototype, multiple paths and alternatives should *intentionally* be implemented for each choice. By reviewing the implementation as a demonstration project, better choices will be made about standardized options for the full implementation effort to follow.

Release planning

The new data warehouse portfolio is a long-term commitment of resources that needs to be carefully planned while exploring alternative scenarios. The architecture of a dimensional warehouse lends itself well to a significant initial effort to get the infrastructure up and running, and then cycling through iterations of new data sources on a regular basis. If such a plan is in place, the pressure to maximize the data loaded into the first production-level release is lessened because of assurances that subsequent data sources will be added routinely.

Recommendation #4: Formalize a planning process in the data warehouse support team, in conjunction with the new data administration and governance bodies, to identify and prioritize data sources to be loaded into the new data warehouse portfolio, emphasizing new release upgrades at least quarterly for the next three to five years.

There will be tremendous pressure on the data warehouse team to maximize the data included in the production warehouse, unless a reasonable and believable plan can be put together that shows how data sources will be added to the warehouse over time. The governance team will prioritize the sources, taking that political pressure off the data warehouse team. The most favorable and workable scenario is to plan an updated release of the warehouse every 3-4 months, with each upgrade adding several data sources. In this manner, most major data sources will be loaded and available within only 2-3 years.

Initial warehousing system

An initial production-level warehouse should be targeted to be implemented as soon as possible, *preferably within a year*. Because of the prerequisite data administration, data governance, and infrastructure issues that must be dealt with, the timeframe for Release 1.0 is unlikely to be less than nine months. Scope should be controlled so that an initial release is available in 9-12 months, with additional data sources added as new releases every few months thereafter.

Recommendation #5: Define and develop a production-level clinical data warehouse and have it operational quickly as an integrated test of all administration, governance, and planning capabilities.

The initial warehouse system is to be scoped by the data governance team, and sequenced according to the release plan. The development team's emphasis is on making all analysis, design, and build activities for the data warehouse ETL as repeatable as possible in support of the aggressive release plan. Repeatability implies an improved level of process maturity.

Organizational maturity

The organizational process maturity of the information technology groups at most new healthcare client organizations typically rests upon some very *ad hoc* process management, where the skills of the individuals doing various jobs are significant, but the documentation and processes for doing those jobs is largely in the heads of the people doing them. As a data warehouses increases the interaction and complexity of the entire health information portfolio, it will put a strain on the individuals in their jobs as the scope of complexity of those jobs increases. A lot of work on information technology process maturity has gone on in industry over the past two decades, with little penetration of those models into the healthcare sector. As an organization's IT environment becomes more complex with large-scale warehousing, it can reduce its long-term risk by beginning to adopt some of these models.

Recommendation #6: Begin the process of identifying, developing, and institutionalizing more mature processes and practices throughout the information technology function of the enterprise.

There are two primary pathways to organizational process maturity, and new client organizations should begin making investments in both: 1) development and engineering maturity, as represented by the SEI Capability Maturity Model – Integration (CMMI), and 2) service and support maturity, as represented by the Information Technology Infrastructure Library (ITIL) and the ISO 20000 standard.

Schedule Implications

Some basic scheduling opportunities and dependency issues among these six recommendations are highlighted in the following prospective work breakdown structure:

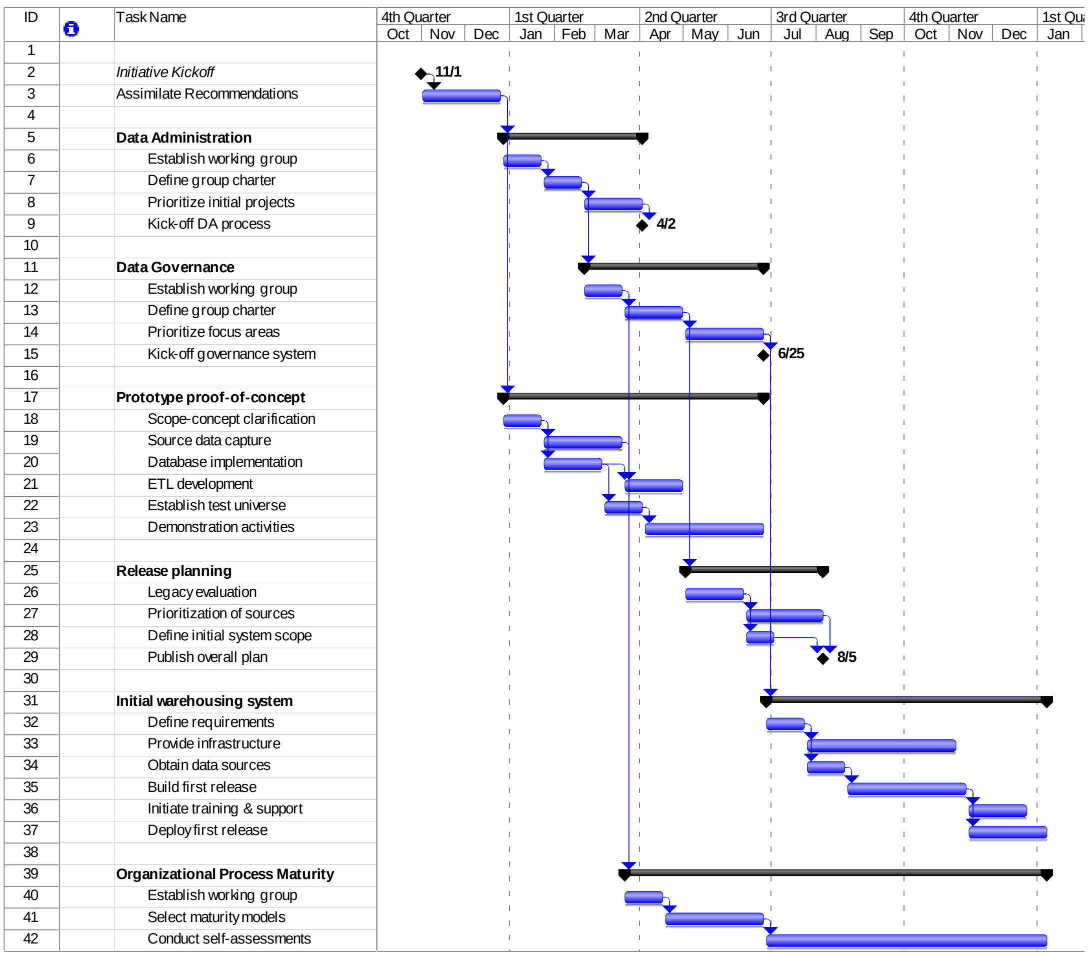

Figure 1 – Prospective Implementation Work Breakdown Structure (WBS)

Table 1 recaps the highlights of the six recommendations, summarizing the general difficulty that can be expected in their implementation as well as a general level of risk associated with the possibility of excluding each from the plan going forward.

Table 1 - Recommendation Timeframes, Difficulty, and Risk

Recommendation	Target Timeframe	Difficulty	Risk if Not Done
Data Administration	First Quarter	Low	Moderate
Data Governance	Second Quarter	High	Critically High
Proof-of-Concept	Second Quarter	Low	High
Release Planning	Third Quarter	Moderate	Moderate
Initial Warehousing System	Fourth Quarter	Moderate	Moderate
Organizational Maturity	Beyond First Year	High	Moderate

The most immediate challenge in moving forward is typically putting the appropriate human resources in place to begin work. The immediate question will be the extent to which the existing organizational data teams can be shifted to work on the new environment while also meeting the needs of supporting any legacy reporting or pre-warehouse environments in the near-term. Beyond that immediate consideration, several resource types will typically be needed moving forward:

- Project Management (.5 FTE, continuous)
- Data Warehouse Architect (.5 FTE, heaviest up front)
- Business & ETL Analysts (1-2 FTE)
- Integration Specialist (.25 FTE)
- Data Base Administrator (.5 FTE, intermittent)
- Universe Designer (.5 FTE, intermittent)
- ETL Developers (2-3 FTE)
- Application Developer (1 FTE intermittent)
- Data Administrators (.5-1 FTE)
- Organizational Change Agent (.5 FTE, focused on Governance)

Based on my experiences at multiple healthcare clients, I recommend preventing this team from growing large. An initial growth in team size often creates the feeling of faster progress, but inevitably involves learning curve and coordination issues that have caused me to scale back the team to an effective core in both places. My previous implementations have been more effective with core integrated teams.