As a business intelligence consulting company, we pride ourselves on being able to deliver on our projects as well as provide good quality content to our readers.

It’s no secret that Business Intelligence* (BI) projects are both time consuming and resource intensive, often suffer from poor communication between the business and IT, and are usually inflexible to changes once development has started. This is due, in large part, to the method by which BI projects are traditionally implemented.

Regardless of the methodology you employ, at the very least, a successful BI iteration requires:

- Business requirements identification
- Data analysis
- Data architecture and modeling
- Data integration (ETL, ELT, data virtualization, etc)
- Front-end development
- Testing and release management

**A DESIGN, DEVELOPMENT AND MANAGEMENT APPROACH TO BI**

Whether you choose to integrate testing with development or employ prototypes and sprints, it doesn’t diminish the fact that design, development and testing are part of the business intelligence lifecycle process. The challenge with the way in which BI projects are usually implemented is that the design and development steps across the architecture aren’t integrated and insulated from direct input by the business.

In addition, since the tools we usually employed for design and development aren’t integrated, both initial development and subsequent changes require a significant level of manual effort and change management. If we want to improve upon the traditional BI development process, we need to start approaching this process as a business driven lifecycle, as well as integrate and automate as much of the development process as possible.

*Note: In this context, BI refers to both the data integration layer as well as the front-end tools for reporting, analyzing and disseminating the information.*

Business intelligence lifecycle management is a design, development and management approach to BI that incorporates business users into the design process and focuses on generating the data models, database objects, data
integration mappings and front-end semantic layers directly from the business user input. The challenge is that traditional BI projects leverage multiple, disparate tools are used throughout the BI architecture to capture requirements, document business rules, perform source system analysis, model the data warehouse and transform, store, report and analyze the target data. In many environments, metadata management applications, data quality tools and advanced analytic applications are also employed. Rarely do these tools share metadata, which makes it challenging to automate development and difficult to determine the impact when changes are required. In addition, business input is indirect, since it must be captured, disseminated and often re-communicated to the development team.

Fortunately, there are a number of business intelligence lifecycle tools in the market that facilitate this approach. With BI lifecycle management tools, users are given the ability to input project requirements, logical entities, relationships, business rules, source attributes, target attributes and business metadata. These act as inputs to the logical model which can be defined and then reviewed by the business.

Once approved, a physical model is generated from the logical model, the database objects are generated automatically from the physical model and the data integration (i.e. ETL/ELT/SOA) mappings are created along with the objects in the persistence layer. Some of the tools on the market also generate the semantic models for front-end tool sets from the metadata captured during logical data modeling. Each step in the process may be reviewed by the business and IT through a workflow process, and dependencies ensure that a change is reflected across the architecture.

THE BUSINESS INTELLIGENCE LIFECYCLE PROCESS

Figure 1: The Business Intelligence Lifecycle Process

The consistency between layers of the architecture is provided by the common metadata layer. Consider it an inventory of BI architecture assets with defined relationships. Since all the attributes, business rules, physical objects, etc, are being
captured, the dependencies can be identified and traced. This enables change impact analysis and facilitates the testing process. The common metadata layer also creates a linkage between all the layers of the architecture so that each layer can be generated dynamically for rapid prototyping, which provides context to the users throughout a BI implementation.

In the example below, if a change is required to one entity in your logical model (in yellow), you could also see what objects would require changes across the entire architecture. In addition, with workflow and version management, changes can be controlled, tracked, prototyped, implemented and rolled-back if necessary. You could also view changes to the entire environment over time. Workflow automation ensures that the changes are approved at each step along the way.

![Figure 2: The Business Intelligence Lifecycle Change Impact Analysis](image)

The business community has long complained that BI takes too long, is too IT focused and doesn’t respond easily to change. If BI is to become more agile, then we need to start approaching the implementation process as an integrated business intelligence lifecycle rather than loosely connected steps. While most business intelligence lifecycle tools are still immature (i.e. only addressing a few areas of the complete process) they provide a glimpse into what’s possible. However, you don’t need to purchase a business intelligence lifecycle tool to get started. BI projects and programs can incrementally add business intelligence lifecycle capabilities into their process, utilizing their existing tools. For example, many ETL applications support mapping definitions that can be generated from MS Excel spreadsheet templates, which can be populated by a data modeling tool.

While this doesn’t address the entire business intelligence lifecycle, it does help you to begin to view the design and development process as a business driven, integrated lifecycle and begin to change the way in which you implement BI. In a subsequent article, I’ll discuss the steps in the BI lifecycle process in more detail as well as how these steps can be integrated across both the design and development of the BI architecture.