



September 2009

Best Practices for BI: What We Have Learned in 20 years

Claudia Imhoff, Ph.D.

Table of Contents

Introduction	3
State of BI Project Management	3
What World-class BI Practitioners Have Learned ..	4
BI Best Practices – Your Project’s Best Friend	6
Summary	11

Introduction

Business Intelligence (BI) is no longer the new kid on the block – we have been building data warehouses, data marts, and the analytical applications to go with these data stores for more than 20 years! And in that time, we have learned a great deal about the right way – and the wrong way – to implement these critical components. We now have sophisticated analytical applications, mature hardware and software, and an educated business community.

There is little question that the critical components for BI success are a set of best practices and an iterative, prototype-based methodology for its implementation. Unfortunately, these last, most critical pieces still elude many organizations and are therefore, the topic of this report.

State of BI Project Management

Let's examine a typical BI project scenario. Many BI project managers make the erroneous assumption that a BI project is just like any other software development initiative – that is, they follow these traditional software development steps:

- Assemble a team of developers
- Gather specific project requirements from business users
- Build a data warehouse or data mart for specific project, department, or function
- Buy reporting or multidimensional tool and create some starter reports or standard queries
- Train the business community on the new capability
- Disassemble the team upon project's completion
- Repeat process when next BI project is requested

So what's wrong with this? Well, it can take too long, cost too much, and may result in BI solutions that miss the mark or may be rejected by the business community. Unfortunately, this disconnected, project-specific mindset promotes the very things we try to eliminate with BI: inconsistency (and therefore unreliability) in data; lack of reuse or sharing of data,

processes, or capabilities; duplication of data integration and data quality processing; and redundancy in data storage, requirements gathering, access and development tools.

Each separate project re-invents the BI “wheel” and the lessons learned from a prior project about the operational systems, their data, and sources of errors, the integration rules used and their intricacies, even the expertise around the presentation tool are lost or not shared with others. The knowledge and skills developed in each project are lost resulting in overall higher costs for the BI environment.

This mentality has resulted in many documented failures of BI. The costs of rework and the missed expectations or opportunities for the business community have derailed many a BI initiative. How can this be avoided?

What World-class BI Practitioners Have Learned

BI practitioners have built and maintained numerous BI environments; they know what works and what doesn't. These come down to a few fundamental lessons:

1. A BI environment consists of *multiple, connected* projects that build upon the previous project's infrastructure and knowledge. The best way to coordinate these efforts is by creating an overarching roadmap or conceptual architecture that governs each and every BI project. This is a high level blueprint for enterprise applications development that highlights the various application integration points, facilitates information sharing across the enterprise and describes the technology environment to Business and IT communities. A governing architecture enables the re-use of BI components and promotes an on-going cost savings.
2. BI projects use an iterative methodology. Unfortunately most IT professional are taught a form of software development lifecycle (SDLC) that is very structured and revolves around nailing down a system's requirements

before any code is developed. This works well for traditional operational types of systems but is really terrible for those environments that have more fluid requirements like BI projects. The bad news is that many BI managers who only know SDLC try to fit a BI project into that methodology, causing severe problems and missed expectations all around. The best approach for BI is a prototyping or iterative methodology where mock-ups of the ultimate application are created, populated with real business data, and then experimented on by the real users of that application – the business community. Prototyping allows the implementers to “see” how the application will be used in a real business setting. Changes are then made to the prototype to better fit its actual utilization, and the project goes through another iteration of testing by the users. Each iteration fine tunes and hones the real requirements until the ultimate production version meets the full requirements of the business community.

3. Requirements go beyond current project. As mentioned each BI project should build upon the foundation created by the previous one(s). A BI team often gets into trouble by trying to fit everything into a single project, expanding the boundaries of the project scope significantly to attempt to satisfy all requirements with a single effort. The scope gets unwieldy and the project loses steam as the timeframe is delayed over and over as new changes and requirements are incorporated. BI is NOT a single shot. Requirements can be postponed or prioritized in subsequent projects as phases of delivery in the overall *program* of BI (more on this later). It is critical that the idea of multiple projects or phases be understood from day one.
4. Use prototypes to gather requirements. Nailing down specific BI requirements by simply interviewing business users is not recommended. Most business users need a bit of help in “visualizing” what a BI application can do for them. Prototypes or mockups of the application have been proven to be the most effective way to fine-tune the

functionality of the application. These models help the team set the expectations of the business community as well. Prototypes save time and money and ultimately lead to a more successful project.

5. A documented solution is mandatory. Unfortunately many BI projects become one time wonders because the team does not document the various aspects of the creation of its data, applications and analytics. The documentation is so much more than just a simple ETL trail of source and target data. It should include all the business rules that came into play, why one source was picked over another one, any errors in processing or in the data itself that were discovered, and so on. Documentation should be about the environment as well as the specific project. For example, the amount of capacity required, growth rate of the data, potential number of users on the system, etc. Documentation should be generated throughout the project – not left to the end of the project.
6. Data quality considerations must be upfront. It is important to note that data quality may not belong solely in a BI project. Certainly the quality of the data being used for analytics can have a significant and detrimental impact on the overall success of the BI project but often data quality initiatives are much more extensive efforts, encompassing not just BI but also operational systems, data stewardship, and the entire set of data used by the enterprise. Therefore, data quality can become a distraction to the overall goal of BI by being far more time and resource consuming than expected for the BI solution. We will discuss this in more detail in the next section as well.

BI Best Practices – Your Project's Best Friend

Now let's examine in more detail the recommended best

practices to help ensure that you create a successful BI environment:

Understanding Differences between a BI Project and a BI Program

A common mistake is to manage BI projects as if they were one-off efforts with no relationship to one another. Most BI project managers find that, after they deliver the promised functionality in their project, the business users begin requesting more and more analytic capabilities. Without forethought on how to manage this continuing “program”, the team can find itself being pulled in many directions with minimal forward progress. This is not only inefficient; it is very frustrating for the business users and BI implementers alike.

The following table compares the characteristics of a BI project to those of a BI program:

	Project-centric	Program-centric
Effort	One time	On-going
Perspective	Specific	Broad
Architecture Requirements	None	Mandatory
Focus	Very short term	Very long term
Benefits of reuse	None	Many
Need for strategy	None	Mandatory

A BI program governs several related projects under its auspices and is set up from the beginning to handle change. It becomes the “command center” for these projects, ensuring reuse of technologies and techniques, appropriate prioritization, funds for overarching activities, appropriate resource availability, etc. The program function guarantees that these projects are linked or associated to an overarching business goal or objective such as “making information available to the organization”, “understanding customers and customer purchasing better”, “making better strategic decisions”,

“controlling operational costs, or “increasing revenues and shareholder value”. As a program, the organization is in a better position to leverage all the activities for the related projects. With no program in place, you run the risk of uncoordinated, chaotic and redundant efforts. These all lead to higher costs of the overall environment.

Finally a BI program must incorporate the mantra that the BI environment should be built for change. Building for change means that the BI implementers have the capability to rebuild areas quickly and efficiently. We all know that change is inevitable; that it cannot be just about rapid deployment of the initial project. It must be about the sustainability and ability to handle changes to the existing environment as well.

Prototype and Iterate for Better Results

Nothing explains the features and functions of a complex environment better than a real working prototype. BI capabilities are not always simple to understand or describe. Seeing one in action – even if it is only uses sample data – creates a shared understanding of the underlying concepts.

And giving the business users a chance to use it and suggest enhancements or determine which requirements can be eliminated greatly increases adoption and acceptance by that audience. A prototype is an excellent way to fine tune the overall scope of the project. As mentioned it has been well documented over the years that BI applications must be built in an iterative fashion and prototyping is the best process to support this type of methodology.

Another advantage of prototyping is that it gives the BI implementers a good idea of the underlying quality of the source data. This can be a great benefit when developing the ETL and data quality processes. It also draws the attention of the business community to data problems that are not caused by the BI environment (BI is only the messenger) and may stimulate the business to pursue correcting these problems where they exist (the

operational systems) before the ETL layer receives them.

Finally the prototype is one of the best mechanisms for documenting the overall features and functions of the final product. The mockup is a fully documented version of the ultimate production application which gives the programmers, DBAs, report creators, and the business users all they need to design, build and use the final version. A recommended approach is to use a workbench such as WhereScape RED where the development team has everything they need for prototyping, ensuring integration with data models, generating code, and support for partitioning of tables.

Ultimately the prototype does the following:

- Reduces costs – The business users design their system not the implementers. This avoids the deadly situation where the implementers are left to “guess” what the users want.
- Shortens the development time – Prototyping eliminates the rework required when guesses turn out to be wrong.
- Fine tunes the ultimate deliverable – Prototyping supports the iterative nature of BI projects perfectly. Business users have a chance to change their minds, add or subtract functionality without causing delays or rework.

Create a Sustainable Management Framework

BI environments must last a long time; decision support never becomes unnecessary or is “retired”. This means that whatever you build today must serve as part of the foundation for future projects. We have discussed documentation in general but now we need to be more detailed. The documentation for a sustainable environment is called metadata – or the data about the data, models, processing, technology, and even the business users.

It contains the technical information about the sources and targets of data, the audit trails of the ETL and data

quality processing, the business rule constraints applied, and so on. It also contains business information such as the aliases for data elements, security requirements, contact information for data stewards, etc.

A metadata repository is a storage mechanism to capture and make accessible this goldmine of information. It must be easily managed, easily understood and support full version backups. Future projects must have the ability to look back to understand why certain steps, processes, and techniques were used. In terms of being able to maintain and sustain a BI environment, metadata is considered the keystone in this overall effort.

Understand Where and When to Invoke Data Quality Processes

As mentioned earlier, data quality is a critical part of any BI initiative but it is not solely the responsibility of the BI team. It is important to understand and *limit* the time spent and effort expended on fixing the data problems that occur in operational systems. The BI team must determine what quality problems they must address in order for the data to be usable in the analytical applications and what data must be fixed at the source.

Perhaps the biggest drain on the team's time and the most common reason for BI applications to miss their deadlines come from the attempts to correct all the data quality problems uncovered during a BI implementation. It is imperative that the project manager set appropriate expectations for the quality of the data accessed in the BI application. Once again a prototype that consists of subsets of real operational data can help with this process.

Experienced consultants know that *fixing the problems where they occur* can greatly reduce the costs, frustrations, and deadline delays of any BI project. And ultimately these corrections improve the overall operational environment as well.

Summary

The past two decades of BI implementations, we have learned a great deal about what to do and – more importantly – what not to do when creating a BI environment. We now know that:

- BI projects must be part of an overarching BI program
- A conceptual BI architecture must be used as the environment's roadmap
- Prototyping is the best and fastest way to gather requirements and garner business community's support
- Documentation is critical for the sustainability and maintainability of the environment
- Data quality initiatives are bigger than BI and should not be buried under BI projects.

The smart BI manager will look for software from companies like WhereScape to help ensure successful projects for ongoing sustainability. Open, iterative methodologies like RED allow for modifications and enhancements to suit the individual company's needs while ensuring adherence to the best practices described here and developed from experience.