Data Warehousing – Change Management In A Challenging Environment

A White Paper by David M Walker

Sponsored by Sybase®
INTRODUCTION

The way in which data warehouse developments and solutions are viewed is changing. The emphasis is less on the technological problems, many of which have been solved, and more on the day-to-day issues of living and working with a data warehouse. These issues include:

- Configuration/Change Management
- Managing and Improving Data Quality
- Engagement with the Enterprise Architecture
- Enhancing Return on Investment

These issues affect both new developments and existing solutions. The solutions to these issues are information based and process driven, i.e. we need information about what is happening and why it is happening in the system in order to drive the processes in both the development and operational environments that manage it.

This white paper investigates the issues that are affecting the data warehouse environment. The paper will also look at how the issues might be addressed and at a tool that can help.

WHAT IS A DATA WAREHOUSING ENVIRONMENT?

We start with a brief description of the data warehousing environment that should be familiar to readers but is included to provide a common point of reference. The data warehouse environment can be described in its most broad sense as the systems and processes put in place to deliver information to business users. The technology can be represented in a diagram such as the one below:

![Figure 1 - Typical Data Warehouse Architecture](image)
In this example the source systems feed a data warehouse that in turn is used to feed either data marts or cubes via Extract, Transform and Load (ETL) processes. Users can run reports that query the data marts or cubes to produce the information they require.

There is also a process architecture of all the things that need to happen in order to allow the warehouse to provide the required service:

![Diagram](data-warehousing-change-management-in-a-challenging-environment.png)

**Figure 2 - Processes associated with a data warehouse**

Here we can see that there is a requirement for on-going support of the architecture, development, operations and data quality processes, each of which will have a lot of metadata (data about data) associated with it and will be repeated many times.

Both these diagrams under-represent the amount of information that is needed for the successful operation of the data warehouse.
As identified in the introduction there are a number of areas that present challenges in a data warehouse:

**Configuration/Change Management**

Configuration and change management is probably the largest single issue facing data warehouse implementation and maintenance. It operates at every level of the organization and is often the ‘elephant in the room;’ we all know it is there and we know that we don’t do enough about it but nobody talks about it. The following examples illustrate the issue:

- An organization has three mainframe servers, each server performs one upgrade every quarter that is planned and tightly controlled with three months for analysis, design and testing. This means that the data warehouse team has to handle a change every month as it is fed from all three of the source systems. The development team has only four weeks to perform its own analysis, design and build for each release.

- An organization has deployed a series of modules from a major ERP vendor to meet the operational business requirements. The business is pro-actively expanding the number of modules in use and has a requirement to have reporting available from ‘day one’ when the new modules go live but also requires all the historical data to be maintained in the data warehouse within a consistent data model.

- The same organization also receives a number of patches and point releases from the vendor with release notes that insist the patch should be applied as soon as possible but do not describe the underlying data model changes that are used as sources in the ETL.

- A small but critical piece of information is maintained in a spreadsheet on a shared drive. Someone decides that they could improve the spreadsheet by re-formating but this impacts the automated load of the data warehouse.

- A server is moved and the new location is firewalled from the data centre in which the data warehouse systems are kept. The ETL processes suddenly can’t see the server and fail.

These are only a few examples of the change management issues that will be familiar to any data warehouse environment.
MANAGING AND IMPROVING DATA QUALITY

Data quality is often considered a major issue with the data warehouse. In general the Garbage In – Garbage Out principle applies and most data warehouses faithfully reproduce the data quality issues in the source system, even acting to amplify some of them. Data quality issues have been around for some time as Charles Babbage noted in 1864:

‘On two occasions I have been asked, - "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" [...] I am not able rightly to comprehend the kind of confusion of ideas that could provoke such a question.’

Examples of common data quality issues might include:

- Discontinuity between source systems: this occurs when two systems are merged in the load process of the data warehouse and assumptions are made as to the use of keys etc. For example at the start of the project two source systems use codes 1 to 5 for the same reason. Later both systems start to use code 6 but for different purposes however the load finds a matching code and continues to load the data even though the join is invalid. This can be fixed in the load process but represents a failure of data quality at the source system that lacks co-ordination of critical data items.

- Attribute reuse: this occurs when a field or attribute is used for one purpose for a period of time then re-used for another purpose. This is common on mainframe technology where space is at a premium and also with off-the-shelf packages that have user definable fields that allow customization. This change of use may go unnoticed and yet fundamentally changes the meaning of the data stored in it. In some cases poorly made changes have allowed two different meanings to occur at the same time.

- Un-enforced referential integrity: Referential integrity is a process whereby data is constrained to a list of valid values held in another table. This can either be manually maintained in the application or enforced by functionality within the database. Where it is manually maintained the system is prone to failure and this leads to data values being entered that are not part of the list of valid values.

- Unstructured data is data that is entered into a long character string field and then parsed by a program to get the information out. For example where an address is held in a single field and the address line fields are parsed based on the commas within the text into a more structured format.
Complex Data is data that inevitably has a high degree of human interaction in it and is thus much more prone to data errors than automated data. For example the widespread use of spreadsheets as a data source always leads to data quality issues. Another example is the entry of names and addresses by call centre operators who often type what they hear (e.g. John Deere, John Deer, John Dear) or use a default value (99% of all our customers are male because it is the default value) or set up a new account because finding the previous one is too difficult or it pays more commission to open a new account.

There are again only a few examples of typical data quality issues that might face the data warehouse environment however it should be noted that they come from the source system and are not in general created by the data warehouse itself.

**ENGAGEMENT WITH THE ENTERPRISE ARCHITECTURE**

The formal definition of Enterprise Architecture is the organizing logic for business processes and IT infrastructure reflecting the integration and standardization requirements of the firm’s operating model (Paul Weill, Director of MIT Center for Information Systems Research). In practice it is how the strategy and architecture team of an organization define the current state of processes and systems, how they define the future state of those processes and systems and the migration path between the current and future states. For a data warehouse this engagement is critical as it addresses the following concerns:

- Which systems to use as master sources? Where is the data created? Which system holds the right or master data? Which systems hold copies or enrich the data? The path from the source systems to the presentation layer and what happens to it along the way is known as data lineage or heritage.

- Which systems are strategic and will still be used and developed for a period of time? Which ones are tactical and likely to be decommissioned before or when the build is complete?

- What systems or process changes can be put in place to improve the systems architecture as a result of the data warehouse build? It is not uncommon for the data warehouse development to ask questions that identify gaps in existing business process that need to be filled

- What needs to be included in the technical architecture/design of the data warehouse to ensure that it is capable of supporting new functionality currently outside the scope of the data warehouse or at least minimize the cost of change associated with adding that functionality?
• What is the down-stream impact of changes in the source system? This is a commonly overlooked issue for project managers and/or system managers that do not have access to information that shows how changes to the system they are responsible for will impact systems that may not be directly connected but are downstream from the source system

• What information technology framework is used by the organization? Following industry and organizational standard frameworks for the deployment of information systems is often used to ensure the appropriate level of re-use and integration

• How do we understand the operational environment? Enterprise architecture can also be used to allow data warehouse developers to understand the data model, backup schedule, batch schedule, interfaces, systems upgrades and technical architecture of the source system.

The resolution to each of these issues relies on being able to quickly and easily see and act upon the enterprise architecture of the organization and the sharing of information in a number of different formats that is useful to those who need it

**Enhancing Return on Investment**

The on-going cost of running a data warehouse, especially in times of economic hardship, is often questioned. It is therefore common to look for ways to improve the return on investment. This can be done in one of two ways: by gaining more financial benefit from the output, or by reducing the cost to manage and maintain the system. Examples of how the return on investment can be improved include:

• Having shorter development lifecycles; i.e. reducing the time from a user’s request to being able to produce the information. To achieve this it is important to be able to capture requirements, analyze sources, design ETL & reports, build code and test the system more effectively.

• Reduced system downtime; system availability ensures that users are able to work with the information. If a system is not being updated because an ETL job is failing or has loaded the wrong information and that information has to be backed out then the system is unavailable which is a direct cost to the users

• Trusted data as a result of good data quality is also a direct return on investment factor. If the data is trusted then the system will be used, if it is not trusted then it will not be used and therefore the investment is wasted.
• Most data warehouses are built with the idea of producing a single version of the truth, this goal is because data warehouse developments are aimed at handling the contradictions in the operational systems. Removing the contradictions removes the cost inherent in managing and explaining the contradictions.

These issues highlight the need for data warehouse projects, enterprise architects and other projects to communicate and share information.

**RISING TO THE CHALLENGE**

The ability to meet these challenges is not about conventional data warehousing technologies (database, extract-transform-load (ETL) tools, reporting tools, etc.) but about the people, processes and supporting tool-set that they have available to them.

As this is a technology-focused white paper it is beyond the scope to look at how to obtain and retain good people and the development of best practice processes but we can look at the tool-set that would be required to support the key roles involved in managing a successful data warehouse environment. The list of tools that would normally be considered would be:

• Project Management Software
  Used to create and manage the project plan and co-ordinate resources across the project

• Source Code Management (SCM) Software
  Used to manage changes in documents, code and other information including the management of versions and releases

• Ticketing System
  Used to track tasks, risks, issues, enhancements, test cases and defects across the project

• Data Modelling Tool
  Used to design the logical data models and to create and manage the physical data models used in the databases

• Office Package
  Used for Documents, Diagrams, Presentations, Spreadsheets, etc.
At this point the basic toolset for a successful project is in place and we can look at how some of the pivotal roles would use the tool-set described above.

- **Enterprise Technical Architects**
  This group of users will often create diagrams and documents that describe the current and future state of the enterprise technical architecture.

- **Project Managers**
  Project Managers will use the project management software to plan their projects and if an enterprise wide planning tool is in place this may create awareness of changes in other systems. The project manager is also responsible for ensuring that the project follows a particular methodology and/or framework for development.

- **Project Technical Architects**
  The person performing this role will need to describe the technical architecture for the project, how it interacts with the enterprise architecture and where the flow of data through the systems. This is normally again done with a range of documents and diagrams. The technical architect may also act to enforce the project methodology or framework on behalf of the project manager.

- **Data Modeller**
  The data modeller will be responsible for the creation of the logical and physical data models of the data warehouse. These are a key component and will be used by business analysts, ETL developers and report builders as a critical source of information. Each of the users will also need to add metadata to this model as they use it. The data model is developed in the data-modelling tool.

- **Business Analysts**
  Business analysts will record the business requirements (normally in documents), analyse the enterprise architecture to find potential source systems, analyse source systems to find the data and define it.

- **ETL Designers**
  The individuals responsible for the detailed design will need to understand the enterprise architecture, the technical architecture of the data warehouse, the source system data model and the data warehouse data model. These designs are often documented in a combination of diagrams and spreadsheets or documents.
• Report Builders  
The people charged with creating the reports will need to be able to use the data warehouse data models and the business requirements to create the correct environment and document it.

• Operations  
The operations team will need to understand the batch scheduling of the ETL and any batch reports, the order in which the ETL and batch reporting is to be run and how that interacts with other batch schedules across the enterprise architecture. They will also need quick access to the history of changes in systems should a batch process fail in order to understand the source of the problem and the remedial/recovery actions needed.

• Data Quality Analysts  
This group of people will spend much of their time interacting with the data models of source and target systems, the ETL used to move the data around and the lineage of the data in an attempt to try and find the source of the discrepancy and a resolution.

The diagram above shows some of the metadata interactions required. For clarity some relationships have been excluded, for example the relationships between data quality and everything else are omitted. This is the start of the documentation processes needed to help successfully run a data warehouse. It is obviously a labour intensive
process and this in itself causes problems. If a process requires too much effort it will not be performed, but if too little information is recorded and analysed the system will quickly fall into disarray.

Furthermore the data created by the process is fragmented with little pieces on information spread all over the place making it difficult to get an integrated view. For example a diagram drawn in an office product loses all the inherent knowledge of the architect that drew it as it does not store the dependencies, relationships and metadata associated with it.

All of this information is metadata – data about data. This is a broader definition than most projects apply because whilst metadata is easy to identify in the ETL (load statistics), Reporting Tool (query statistics) and database the effort required to find and integrate other sources is deemed too expensive or just not possible

**SIMPLIFYING THE PROCESS**

The issues discussed above leave most projects in a difficult position; spending the time trying to produce and maintain quality metadata and consequently a quality system costs money and the benefits are hard to quantify Not doing it means the system goes into terminal decay from the outset which always ends up costing more.

> Quality is free. It’s not a gift, but it is free. What costs money is doing unquality things – all the actions that involve not doing jobs right the first time. (Philip B Crosby, Quality Is Free, 1979)

So what is needed is a way to simplify the management of all the metadata that exists outside the ETL and Reporting tools and the database itself and integrate with those tools.

To that end we will look at how using PowerDesigner15 might help this process. Since no project has the luxury of a green field site we will assume that it is being introduced into a large organisation with an enterprise architecture/strategy team and a new Business Intelligence team that is just starting work on a replacement data warehouse system. The organisation has the usual array of multiple source systems and reporting systems and the usual pressures on budget and time.

- The Technical Architect

  Faced with a seemingly impossible amount of information to collate and a project manager who wants to know what tasks to prioritize and put on the plan the architect sits down and quickly adds his vision of the future system technical architecture and the current major systems that will act as sources within the organisation.
Already we have the start of a current technical architecture for existing systems and a future state for the data warehousing solution. These are all recorded as enterprise architecture models in PowerDesigner.

- The Business Analyst

Whilst the technical architect has been trying to produce a roadmap for the new system the Business Analyst has been looking at the current state of business intelligence and trying to document the requirements going forward. The Business Analyst holds a series of workshops where two types of information are forthcoming.

The first is the existing flows of information, how data is extracted from systems, manipulated (often in spreadsheets) and then passed to another individual who also manipulates the data before loading it back up into a system that is used for reporting. This information is recorded into PowerDesigner as a series of Information Liquidity Models.

The second set of information is the requirements of the new data warehouse. These come from many sources, some of which are technical (e.g. performance, access, security) and some of which are informational (what data is required, what business rules should be applied to the data, etc.) These are all captured in PowerDesigners' Business Requirements functionality.

- The Enterprise Architect

The Project Technical Architect invites the Enterprise Architect to share the systems strategy going forward. At the meeting the enterprise architect provides a number of diagrams he has created in a drawing package about the future state. These are imported into PowerDesigner and notes added about timescales etc. The Enterprise Architect is also able to identify some key stakeholders of the existing systems that have already been entered into PowerDesigner by the Technical Architect.

When the Enterprise Architect gets back to his desk he uses the URL that the Technical Architect provided to access the system. Since the new enterprise architecture has already been imported he is able to share it with a number of other projects and mails out the URL providing quick and easy access to other users.

The next time that the architecture has to be updated he creates a new version of the enterprise architecture and replaces the
imported diagrams with ones drawn in PowerDesigner itself. All the projects that access PowerDesigner via the web also see the new diagrams as soon as they are released. The new versions also support additional metadata that was not documented in the previous diagrams.

- The Project Manager

The Technical Architect and the Business Analyst are now in a position to sit down with the Project Manager and start to describe the development stages of the project for the plan. Already it has become clear that some of the source systems do not contain all the information required and will require additional analysis. It has also been noticed that one source system that had initially looked like it would take a long time to analyse will not be needed and thus can be removed from the plan.

The project manager is also concerned about the sponsors’ team engagement with the project so he has one of the business analysts enter the organisation chart into the enterprise architecture and links the individuals to the systems. He discovers that one of the sponsors’ teams which hasn’t been included yet are the owners of a critical system. Using the same information he is able to see which business teams have contributed to the requirements gathering process and can follow that up.

- The Data Modeller

This individual now has access to the current version of proposed technical architecture and the business requirements and can start creating the conceptual, logical and physical data models for the main data warehouse. If the business requirements or technical architecture change then it is quick and easy to assess the impact and inform the project manager of any subsequent delays. Each version of the data models can be tied to a version of the architecture and of the requirements.

Once the data warehouse model is drafted it is possible to automatically generate the star schemas and cubes for the data marts. These can then be refined to meet specific requirements.

- The Business Analyst

Although already mentioned the Business Analyst now starts to do source system analysis of the major systems. The first step is to reverse engineer the data models from the source systems and do some profiling of the data. PowerDesigner is used to reverse engineer the data model before the profiling tool results are added as additional metadata.
The organisation has deployed an ERP package such as SAP and therefore by using an extension to PowerDesigner it has been able to generate a complete view of the system and the business rules. However just as the analysis approaches completeness there is a major patch to the ERP system. Fortunately by importing the new ERP metadata and comparing it to that already in the system it is clear that the patch has no impact on the analysis.

The business analyst now moves on to defining the mappings from the source system to the target data warehouse data model. Both are already held within PowerDesigner so a Liquidity Model provides the quickest method for defining the ETL mappings.

The Business Analyst can also use PowerDesigner to record all the business processes, not only is this useful for analysis of the system but provides a source of information for understanding the impact in changes to the process, not only for the data warehouse project but anyone who is responsible for the management and maintenance of systems.

- The Project Manager

The project manager is pleased with progress, especially if he is almost on schedule and calls a review meeting with the business users, technical architect and business analysts.

At the meeting the architecture, data mart data models and business requirements are reviewed and the business users promise to go back and review the business rules in the ETL mappings. A few days later after using the web based interface version 1.0 is signed off and handed over to the ETL developers.

Inevitably a few days after sign off three additional requirements are raised and added to the system. The impact analysis functionality allows a quick and easy assessment and two of the three additional requirements are agreed and rolled into Version 1.1, whilst the third (major) requirement is put off to the version 2.0 release. The sponsor had wanted the information produced by the third requirement but was realistic when he saw the impact and could therefore get a cost/benefit analysis.

As the system gains more information it also becomes the project managers ‘early warning system’. This is because as Technical Architectures, Business Processes or Operational Systems are changed and these changes reflected in the PowerDesigner repository the impact analysis will quickly show...
the impact of the change to the data warehouse build. Other project managers often quickly use this functionality too!

- **The ETL Developer**

  With a versioned definition the ETL developer has a clear goal and timescales with which to work against. This allows them to plan the best order in which to develop the mappings so as to deliver to the testing team functional sets of data in the data mart ready for testing.

  As a result of the business analysts keeping the source system data models up to date it is also possible for the ETL developer to keep track of changes in the source system and accommodate them quickly and easily into the ETL being developed.

- **The Test Manager**

  The test manager can now pull together end to end test plans that allow him to validate that there is a requirement, that the source system is as envisaged when it was analysed, that the mapping matches the designed mapping finally that the requirement is met.

  All the metadata about this and the liquidity and impact analysis models are held in the single PowerDesigner repository allowing easy access and analysis of any problems that arise in testing, which in turn completes the feedback loop to the developers to fix the bugs.

  Finally when all the mappings appear correct any outstanding data quality issues can be handed over to the data quality analyst for further examination

- **The Data Quality Analyst**

  The data quality analyst is dealing with issues from three sources:

  Profiling – using a data profiling tool to discover issues in the source systems and requesting changes back to the owners of these systems. These issues are flagged in PowerDesigner metadata by the Analyst which allows the ETL developer to take special care when developing the ETL that reads this data

  Automated Checks & Tracking – by using data quality queries that are run regularly and on-mass trends in the data quality can be spotted. These trends can be tracked from the source using the lineage functionality and the causes identified and rectified.
User Identified Issues – As already suggested groups such as business users and testers will identify data quality issues that will have to be resolved back to the point of origin and the remedy implemented. PowerDesigner once again provides a quick trackback analysis process and access to metadata about source systems and changes in source systems.

- The Operations Manager

The Operations Manager can make use of all the metadata collected during the development process to help maintain the smooth operation of the system once it is in production. The ability to look at which systems are involved in any batch process ensures that failures due to systems outage or changes in source system can quickly and easily be identified.

It also allows the Data Warehouse Operations Manager to look for batch processes that can be pensioned off, especially where metadata such as query usage is imported into the PowerDesigner Repository, as maintaining and running ETL and Reports that are never used is an expensive process.

Finally it allows the operations manager to analyse changes and assess the impact before allowing changes to take place that will improve the overall delivery of IT services to the business.

This example has demonstrated that a wide-scale ‘viral’ deployment of PowerDesigner started from within a data warehouse project can have significant impact on the development times, the quality of the final delivery and the operational environment in which it functions all of which reduce cost and improve the quality and sustainability of the delivered solution.

In addition to PowerDesigner this type of implementation would also require a ticketing system for tracking risks and issues as well as a version control system. The major success factor however is the people and their willingness to adopt suitable processes to ensure the success of the project.
SYBASE POWERDESIGNER FUNCTIONALITY

Whilst we have looked at how Sybase PowerDesigner might help a project it is also useful to describe the set of functionality that is specifically used. Many of the features that we will now discuss have existed in previous versions however the latest release brings them together with new features that create a compelling metadata management solution for a data warehouse project and significantly reduces the overhead of data warehouse management.

CORE FUNCTIONALITY

PowerDesigner has a core of modelling functionality that allows just about any type of diagram or document required for the data warehouse to be created in a single environment

- Enterprise Architecture Models

Enterprise Architecture Models create a representation of the systems, people, processes and information flows within an organization. They are commonly used to document the current state of systems and to plan future states. Existing diagrams, for example in Visio, can be imported and then enriched by adding the metadata to each of the objects.

There are a significant number of different enterprise architecture models supported including

- Process Maps
- Organisation Charts
- Business Communication Diagrams
- City Planning Diagrams
- Service Oriented Diagrams
- Application Architecture Diagrams
- Technology Infrastructure Diagrams

These various formats allow every facet of the enterprise architecture to be documented.

- Business Process Models

Power Designer allows business processes and process hierarchies to be captured in a number of process languages. This allows business analysts to document the business processes and workflows that involve information use, production and consumption as well as to understand which systems are critical to which business processes and therefore
to chose the most appropriate source systems for the data warehouse

• Requirements Models

This part of the tool allows the documentation of requirements, the creation of a traceability matrix and also user allocation matrices. Requirements themselves ensure a clear understanding of the business goals, strategies and tactics that drive the data warehouse initial design and ongoing change. Centralizing this knowledge in the requirements model removes the need for a large number of documents and spreadsheets that are manually maintained. It also means that standardised templates can be used across the organisation

Frameworks Support

If your organisation uses a framework such as the Open Group’s TOGAF or the Zachman Framework this can also be created in PowerDesigner and used as a reference point for all development and architecture. The framework can include all the diagrams and supporting information required.

• Information Liquidity Models

These are models that can be used to show the movement of data, including the design of ETL and the tracking of data quality issues through the system. This is a powerful tool for designers, developers and operations staff in the management of ETL jobs and as it allows quick and easy tracking of where and what is involved in a specific job or set of jobs. Similarly it allows a data quality analyst the opportunity to track individual data quality issues back to their source.

• Data Modelling

The bedrock of the tool, it’s data modelling capability has been extended to fully support conceptual, logical and physical data models in a number of data modelling conventions allowing complete understanding of the data structures. This is combined with the ability to reverse engineer existing systems to quickly improve the understanding of sources and to forward engineer the data warehouse data models for just about any platform to make it a data model one-stop shop.

Using one tool for the ongoing documentation and maintenance of all sources, data warehouse and data mart structures increases consistency of standards use, making it easy to reduce inconsistencies in information definitions and descriptions throughout the enterprise. It also now supports the
semi-automatic creation of data cube models from existing data models that again can be used to speed up the development process

**INTEGRATION AND IMPLEMENTATION FUNCTIONALITY**

As important as all of the above are some features of PowerDesigner that separate it from the pack. In addition to the rich core functionality is the integration and implementation functionality.

- **Version Control**

  PowerDesigner15 has significant built in version control that has the ability to create versions and branches of any model or set of models right down the object level i.e. a table or an entire model can be versioned. This allows architects for example to create current and future state enterprise architecture on separate branches and to version each of these branches as both the current and target architecture change. This provides tracking, gap analysis and change management to understand how the organisation is evolving and plan accordingly. This feature can also be used for any other model in the same way.

- **Centralised and De-Centralised Working**

  PowerDesigner allows users to work either with a central repository or as a standalone product. This means that users can work on their own models away from the central server and until they are ready to publish them, and then to bring them in to the central repository for sharing. This overcomes a common problem in some large projects where developers are unwilling to share until they have a usable version but quickly need the information available when it is published.

- **Web Interface**

  Whilst the tool is a desktop based system there is also a web interface to the repository. This means that the knowledge workers who need to access and modify the system will use the desktop client while for those that need access to a rapid wide deployment of the information across the organisation it is possible to grant access via the web interface. This also provides the ability to get direct access to the latest information at a very low cost across the organisation and to share information with suppliers/system integrators via restricted VPN access to the web server.
• **Extensibility**

Power Designer has multiple ways to quickly and easily extend functionality. PowerDesigner has a graphical user interface used to easily extend the meta-model, adding new details to existing model elements and adding custom forms to make user entry of added details natural and easy. Integrated VBScript capabilities allow for fast and easy extensions to the model check rules to add automatic enforcement of data governance and standards as well as easy automation of common tasks. PowerDesigner also has a COM interface that allows developers to extend the product to talk to other systems such as the Metadata created within the ETL tool or statistics from the reporting tool. In effect it is possible to extend PowerDesigner to include not only the ETL design but also record when it was last run and the outcome (success/failure etc.) of that run. It could also show how often a report was being run and if it was no longer being used schedule it for retirement. These capabilities consequently bring about maintenance cost reductions.

• **Integrated Metadata**

As all of these models share a common meta-model there is great re-use. If the Enterprise Architecture defines a system it is possible to drill down to the underlying technology diagrams and from there to the data model. This can then be followed through to the ETL to the data warehouse model and consequently to the reports that make use of the data. This allows source system administrators to look at the impact of changes or assess the impact of business process changes.

**Impact Analysis**

PowerDesigner has impact analysis reporting and impact analysis diagrams that allow for easy visualization and sharing of the impact of any proposed change. Upstream “lineage” and downstream “Impact” are all shown together to ensure that a change proposal is consistent with the business requirements and conceptual data definitions that guide it, as well as ensuring that the changes cascading downstream from the accepted change will be fully taken into account.

• **Design-time Change Management**

Once a change proposal is accepted, all impacted artefacts are reported and change can be easily cascaded through PowerDesigner by regenerating models that were generated from each other before, or by following dependency links through to impacted objects and changing them in place. These
changes then drive the creation of new code and change DDL and DML to impact the source and target systems together in a choreographed manner.

These features are not only useful in their own right, they make the whole bigger than the sum of its parts.

**EXTENDING PowerDesigner**

Whilst all that has been discussed above is impressive, several vendors have taken it further by adding functionality that extends PowerDesigner even further.

- **Silwood**
  
  Silwood is a tool that enables users to explore, document and visualise data structures in their large Enterprise Application Packages (EAP) such as SAP, JDEdwards, PeopleSoft and more. This ability to look at the site-specific implementation of the EAP and export it into PowerDesigner can significantly reduce the cost overhead of analysis or EAP systems for the data warehouse solution.

- **Meta Data Integration Tools**

  It is possible, either with local bespoke code or packages, to export the run time metadata and import this information into the system to provide and even more complete view of the process
IMPLEMENTATION TIPS

Whilst PowerDesigner provides a wealth of functionality it is important to remember that it is the people and processes that will make it work. To that end there are some simple deployment techniques that should be remembered.

- Make it available to everyone

Either via the desktop or the web-based interface the system should be available to as wide an audience as possible. Furthermore if someone wants to use PowerDesigner for another project then encourage them to do so. This is because when that project delivers a system the metadata required for the data warehouse about the new system will already have been created. The time and analysis benefit easily out-weighs any desktop licence cost involved in providing the software to the project.

Publishing the data via the web allows users to understand the data quality and load issues. This web-based metadata should be integrated into the users front-end solution so that it is an integral part of the way in which reports are used.

- Tutor & Evangelist

Appoint a Tutor and Evangelist for PowerDesigner. It is not sufficient to mandate its use for the project. Project Managers should be aware that an individual within the team is there to encourage users to make use of the tool and espouse it's benefits (the evangelist) and at the same time be willing to spend time with users to show them how to do things or improve the way that they work with the tool.

Once again this technique is for the far-sighted project manager as there is always a perceived cost to doing this. However the benefits over the lifetime of the project of making full use of the functionality of the tool are so significant that this should be considered an investment rather than a cost.
• PowerDesigner ‘on Rails’

PowerDesigner is an incredibly powerful tool and can support multiple process languages, modelling formats, etc. Whilst this is a great advantage it can also create chaos. A similar problem exists in programming languages such as Ruby. In the case of Ruby there is a best practice way of working called ‘Ruby On Rails’ which favours using convention over configuration for software development. A similar practice should be adopted for the implementation of PowerDesigner within the organisation. To this end a single language or modelling format should be used across the organisation for any particular diagram type.

If the organisation has the tutor & evangelist in place then this standardisation is something that they can undertake, if not, it normally falls to the technical architect to perform.

In a similar vein it is also important that projects do not proscribe the use of every model type during the project, instead they should concentrate on the models that are sufficient for successful implementation
CONCLUSIONS

In this paper we have looked at the following:

- Why data warehouse projects have become more about change than about the basic technology tools used to build a data warehouse
- The sorts of information that projects have to gather, record and manage as they evolve in order to be successful in delivering the business intelligence solution
- How people and processes are more important than tools in the development of a data warehouse
- The way in which deploying PowerDesigner into a project can significantly reduce costs and timescales in a realistic project scenario
- How PowerDesigner as a tool has matured with the current release to become one of the most comprehensive tools to support the people and processes involved in a data warehouse
- That using all the components of PowerDesigner is worth more in cost and time savings than the sum of its parts
- The necessity of ensuring that as a tool it is not dumped on a project and expected to provide a ‘silver bullet’, instead its deployment must be nurtured to ensure that the benefits and potential are fully achieved

ABOUT THE AUTHOR

David Walker is a principal consultant with Data Management & Warehousing, a consultancy that specialises in helping organisations in the successful implementation of large-scale data management and data warehousing projects.

David has over 15 years of experience of technical architecture and project management at the highest level in organisations around the world.

David can be contacted via e-mail at davidw@datamgmt.com or on the telephone at +44 7990 594 372.