Paper 141-27

Data Warehousing - What's It All About? Learning to Walk in Seven League Boots
Steve Morton, Applied System Knowledge Ltd., Henley-on-Thames, UK

ABSTRACT

Data warehousing has been going on for several years now; it is widely done, but not always well done. So what's it all about anyway? It is widely agreed that successful data warehouses are built in manageable steps, incrementally - and in today's business climate any long-term project is doomed if it can't deliver business value along the way. However, if you are new to data warehousing how do you know what are the right steps to take when starting that journey? And what are the landmarks along the way?

By learning from others' experiences - good and bad - we can all benefit, with more success in our projects. Even the experienced practitioner will often find useful tips from someone else's successes. This paper examines some of the factors that help make data warehouses work (and some that can trip you up!), including:

- Can you call your project "data warehouse"?
- Maintaining a business focus - prioritizing for value
- Incremental building - how big are your steps?
- The relationship between data marts and the enterprise data warehouse
- Getting the most out of your investment in metadata
- Dealing with data quality issues

INTRODUCTION

"Seven League Boots"? These are mythical, magical footwear that allow the user to travel 7 leagues in a single step (or 21 miles using modern distances). In the Discworld® novels, Terry Pratchett reminds us that great care is required when wearing a device that can cause the right foot to land 21 miles away from the left foot.

The steps in building a data warehouse a lot like that too! Somehow no one ever wants you to take small steps.

So how can you avoid being "over stretched"? Start by recognizing the potential problem – and read on!

This paper is intended for team leaders and project managers considering their first data warehouse project, with an emphasis on practical project issues.

IS YOUR PROJECT A "DATA WAREHOUSE"?

You may wonder if your project is a data warehouse – after all, it is rare that a business manager comes along saying "we need a data warehouse". They are most likely to say something like "we need to analyze customer activity" or "we want to predict cancellations" or "our sales reporting takes far too long".

You may decide that the answer involves a data warehouse, but you are delivering a solution to a business need!

DOES IT MATTER?

Does it matter what you call your project? Often the choice is political rather than technical – either you <u>must</u> call it data warehouse to get necessary support, or (increasingly often now) you <u>cannot</u> call it data warehouse because some objections exist. Common objections are:-

- There is already a 'data warehouse' (even if it's not accessible to the business users!).

- A previous data warehouse project failed, or took a very long time to deliver results
- Stories of failed projects elsewhere, or of runaway costs, have led to a reaction ("we don't want that to happen here")

Many "creative alternatives" exist, and you can probably think of more. Ones I have used include:-

- Analytic and Reporting Data Store
- Detailed Extract Database
- Integrated Detail Layer
- Common History Datastore

Whatever you do call it, if your project has the <u>characteristics</u> of a data warehouse then treat it like one!

Why does that matter? Well, some specific problems and issues that are rare for other types of project affect almost all data warehouse projects. If you are prepared for these, you can limit the adverse effects.

RECOGNIZE THE SIGNS

Since the original business requirement is usually described in reporting or application terms, you will need to be on the lookout for the telltale signs.

A single application data store is very specific, meeting the needs of a small range of users. Though it might integrate data from a few source tables, perhaps a couple of operational systems, its treatment of data is narrow and focussed only on the specific application. It may implement business rules that are only used within a single department, and supports a single historical perspective. Data have often been summarized for a specific purpose. This is often recognized as a "data mart".

In contrast, a data warehouse is more general-purpose. It integrates and prepares data for a range of uses, often across multiple departments. Data are sourced from many operational systems, integrated using business rules that are organization-wide. Data can support a range of historical perspectives, and are available at a fine level of detail (often the individual transactions). The same base data can support reporting, analysis, mining, OLAP and applications by delivering appropriate marts for each.

DEVELOPING THE DATA WAREHOUSE

The days when a corporate IT department would fund development of a data warehouse as an infrastructure project are long behind us. Their legacy is a few large-scale monolithic databases, and a feeling that data warehouse means "big and expensive".

This full-scale, big-bang approach is rarely seen now.

It is important that projects deliver to the business quickly, and without extended development time. So it's tempting to jump straight to the data mart; but this is very shortsighted.

With a little forethought you can solve the immediate need <u>and</u> prepare for the future at the same time.

THE TROUBLE WITH INDEPENDENT DATA MARTS

The first time you build a data mart, everything seems simple. A couple of data extracts, join some tables, summarize and deliver to the users.

But consider the effects once you have several of these (see Figure 1).

Multiple extracts are inefficient and rapidly become uncoordinated; it is almost impossible to ensure consistent use of business rules leading to 'multiple versions of the truth'; and this is even worse if different teams are responsible for each part.
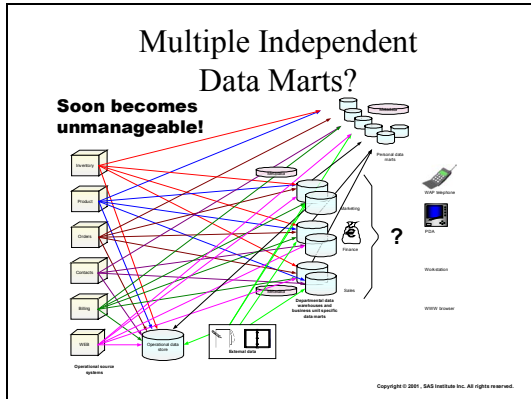

Figure 1

It is better to have a consistent approach. This is what led to the Enterprise data warehouse of the mid-90's (see Figure 2). In this case all data marts are dependent, derived from a central detail data warehouse.
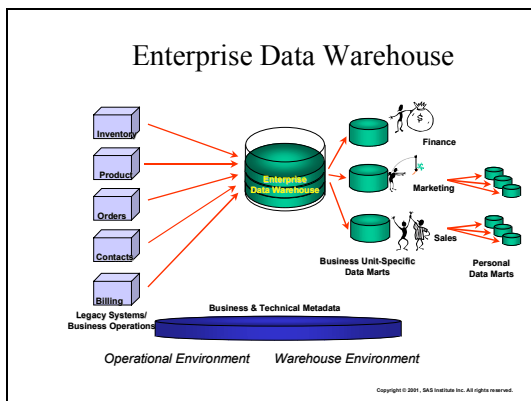

Figure 2

A more recent concept introduced in the late-90's is the "bus architecture" proposed by Ralph Kimball.

Here the coordination is provided by common *process* rather than a central data store, but the basis of agreed business rules, dimensions and sharing of data across the organization is much the same (see Figure 3).

Agreeing common standards and definitions is mandatory to this approach, and is what distinguishes this from the "independent marts" tangle. In my experience gaining this agreement is the hardest part anyway - not the technology or the style of architecture.
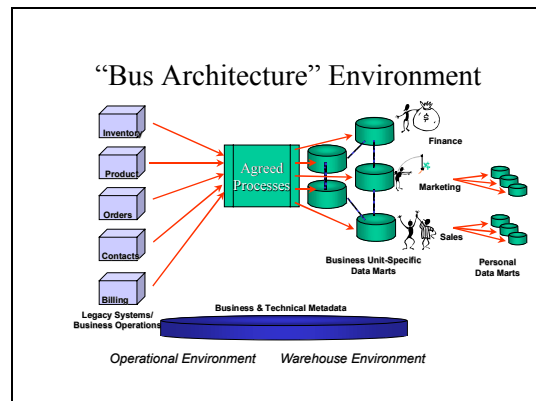

Figure 3

EVOLVING THE DATA WAREHOUSE ENVIRONMENT

Clearly achieving all this in one project is a huge step! Even a single part of it is potentially a large step.

A safer approach is to build in stages, based on an overall vision and plan for the longer term (see Figures 4 through 8).
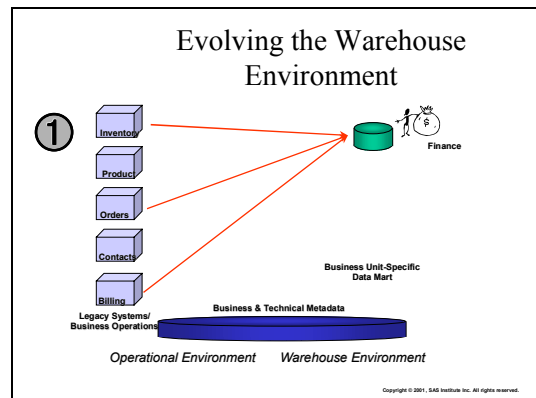

Figure 4

Going straight to a mart alone is tempting – but creating the first part of the long-term data warehouse is a better start if you can begin with that.
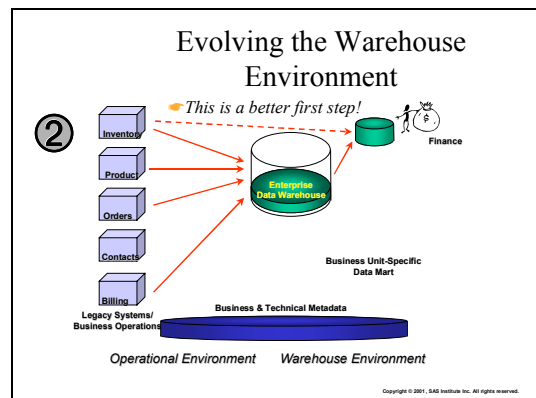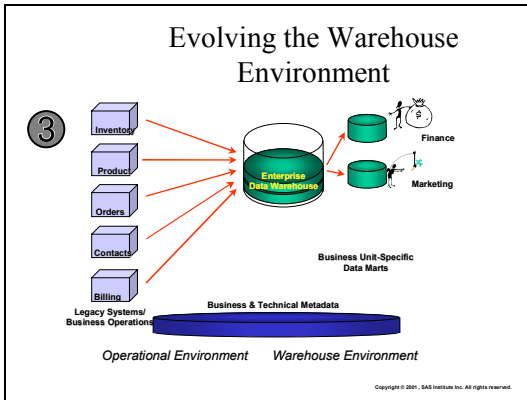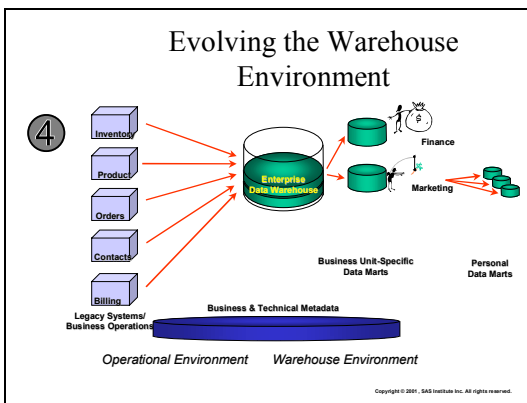

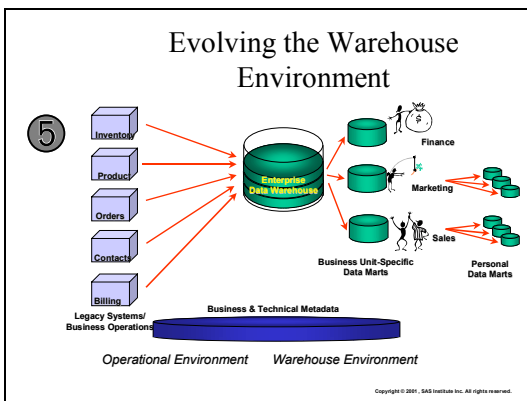**Figure 5**

Figure 6

Figure 7

Figure 8

Evolution for a "bus architecture" is much the same, but with common agreed processes replacing the central data store of the enterprise data warehouse.

The key factor is growth through incremental building. Each step provides more business value – delivering new data subjects, new marts or reports, new ways to exploit the data resource.

The incremental approach is promoted by SAS Institute through its Rapid Warehousing Methodology (RWM), as well as being supported by industry 'gurus' including Bill Inmon and Ralph Kimball (see References).

MANAGING THE STEPS – HOW BIG IS 'TOO BIG'?

When does a step become "too big"? That depends on your data and business rules – but it is not simply a question of gigabytes. Although large data sizes may be a factor – because they take a long time to process and a lot a disk space to store – a greater factor in many projects is complexity.

The more complex are your data, transformations and calculations the more work and risk these add to the project. Especially if data sources have never been combined before, you should expect surprises along the way.

From my own experiences, I have a few "rules of thumb". When a project hits one or more of these, I go looking for trouble – on the basis that I'd rather find trouble before it finds me!

Those alarm points are:

- More than 15 source data tables

- More than 2 source application systems or databases

- Multiple complex transformations

- Many additive measures (more than 3 or 4 'facts')

- Drill-down reporting with a complex formula

None of these alone is going to make the project impossible – but each one adds to the risks, and will take time.

Why those numbers? Well, it just seems to work out that way. Each interaction requires business rules – and everyone will tell you their business rules are simple and well understood. In practice this is not usually true!

Discovering the gaps in understanding and filling them will take time. Usually some gaps become apparent only after data have been loaded and are being reviewed by users – by which time re-work of programs and re-loading of data will be needed. Allow time to do this.

DEALING WITH 'TOO BIG' STEPS

We need strategies to help us cope with 'big steps'.

If possible, try to get the project broken down into smaller sections, delivering each one in turn. Limit the data subjects – delay some data to a second build – or delay some of the more difficult reporting to a second stage. Often requirements change anyway – especially if it is the first data warehouse the business users have seen.

Prioritize based on business value – but at the same time, recognize that the highest value item may have the highest project risk. Business imperatives may require you to address a single large high-risk implementation – if so, you will need considerable commitment from your sponsor!

If you can't reduce the deliverables, look for ways to sectionalize the project itself. This will give you confidence, and provide real milestones for progress. The best milestones are based on data – either related to source systems processed, or data warehouse subjects loaded.

For example, if you are building several data subjects from multiple data sources, write the process for one subject or source completely first and load that data. This lets you practice going live with a less complex set of processes, so you can use lessons learned to adapt what you do for the remaining data.

DELIVER VALUE ALONG THE WAY

Look for useful deliverables that may be ready before the entire project is done. If you can provide some part of the total then you may reduce pressure to show results.

However, remember that once you have some live users you have a production data warehouse. If you make this move, ensure that you have support arrangements in place or else your

ongoing development will be impacted.

Better is to restrict use of an interim deliverable to your user-testing group – a small number of expert users who will give feedback, and eventually become part of your support network. In this way you can gain the benefits of an early deliverable without suffering too much impact.

KEY ROLES

By far the most important role within any data warehouse project is the *business sponsor* (sometimes called the Executive Sponsor). This is the person with business need, budget and vision to get the project done.

You will also need IT sponsorship and support - a major risk occurs if a project is business-led without IT, since the team will have difficulty resolving technical questions.

WHO IS THE SPONSOR?

In the early days you may have to start with a department head as your sponsor anyway. Until the concept is proven it may be difficult – or unwise – to have very high-level visibility. In this case consider what will happen later, since you will need to broaden support over time. Your organization may work well with a 'federation' of department level sponsors, or you may have to look for support to come from higher management.

SPONSORSHIP AND POLITICS

Your project sponsor is your primary champion within the organization, particularly for handling high-level politics.

Yours will not be the only project competing for time and attention – so the ability to influence and get things done is the most important attribute in a sponsor. Clearly it helps if the sponsor is at the highest level within the organization; but sometimes an energetic sponsor one level down is more effective than a board-level sponsor who just wants to sign the check and leaves you to get things done on your own!

You will need to engage long-term sponsorship. Unlike a simple application – designed, built, deployed and used – any data warehouse needs to grow and be maintained. Even if no new data sources are added after a certain time, there are still changing business conditions to affect how the data warehouse is used and exploited. These lead to changes in the data warehouse over time.

PROJECT MANAGER AND WAREHOUSE ARCHITECT

After the sponsorship, the two most important roles are the *project manager* and the *warehouse architect*.

Why two separate roles? Because the two jobs carry very different responsibilities.

The project manager is responsible for the schedule, resources (both people and facilities) and organization. That includes ensuring that management and the team communicates effectively - both internally to the project and with the organization as a whole.

The warehouse architect is responsible for the technical vision and design, ensuring the implementation goes to plan and solving technical problems that arise. This person may also be called *technical leader*.

Can you combine the two? It takes exceptional discipline to deal with both aspects equally well. In the busy environment of a data warehouse project this is a challenge – my advice is do not try to combine if at all possible.

If you have no choice but to do both jobs, be very careful to allocate separate time to both activities and remember which role you are performing in each meeting or discussion!

RUNNING THE DATA WAREHOUSE

A key point in life of the data warehouse is changing from development to production running.

When running in production, jobs need to be as automatic as possible. The data warehouse team should only get involved in these jobs if something fails in an unpredictable way – otherwise routine jobs should be the responsibility of the operations team. I recall one data warehouse that I reviewed where the data warehouse team itself ran and checked every job, every day; as a result they had less and less time to consider new requirements.

If the jobs do require a personal check before the load is considered OK, then consider automatically emailing the logs to the person checking, with an indication of success or failure in the subject line. SAS has some really useful features to help you do this! Then checking should take only a few minutes each day.

PUBLISHING THE DATA

"Publishing" is a term much used by Ralph Kimball to describe what a data warehouse does – and it is an excellent word to describe this. The parallels with magazine or news publishing are very strong.

In the print world, a responsible publisher can be trusted to provide reliable information, and to check their facts before publication. Whilst they cannot guarantee 100% perfection, they do assure a level of quality.

For the data warehouse we must check our facts too!

We have learned to read newspapers and judge how reliable a story is. Words from anonymous 'sources close to' a person quoted in a tabloid gossip column are usually less reliable than a direct quotation in, say, the Wall Street Journal or Financial Times. However, we may still read the gossip, even if we don't fully believe every word of it!

In the same way, users of data warehouse content can still get value from less-than-100%-accurate data. But is important for them to know what level of trust to put in it.

A news story is only as good as its' source – and the same is true of a data warehouse. When publishing in our data warehouse we want to aim for a high degree of trust from our 'readers', and one way we can help this is to be very clear about where the source data come from. This is an important piece of *metadata* – as is any other assessment of quality we can make.

PUBLISHING METADATA

During a data warehouse project a large amount of metadata is gathered and created. This includes:

- Information about source systems and the meaning of data from them

- Business rules to be applied when reading, transforming, joining data and calculating derived values

- Business rules used in reporting and analysis

- Quality assessment – how complete is any source, how well validated, and therefore how reliable; also any dynamic quality measure0s, such as error rate from the load processes

- Who to call if questions arise (not just the helpdesk – identify the 'owner' or expert on that data too)

This needs to be published as well as the data.

Of course we make direct use of this within the data warehouse team when writing and testing data management processes – but a far greater value comes from making this information available to users of the data. Your users will be better informed, and you have a much greater chance to find out when something is wrong or has changed when it is widely visible.

Some metadata naturally get published anyway – for example, OLAP navigation rules (dimensions, hierarchies and analysis measures). For the rest, consider how to make information about the data available alongside the data itself.

Common methods are to use Windows help files or html pages – both are searchable (an important requirement for metadata) and may be used as 'context sensitive' information for data users. Both can be generated programmatically from formalized metadata (more on this below, see 'Metadata and Code').

VALIDATING METADATA

Just as data must be validated, metadata must also be validated before publication. Whilst the rules that govern how programs work must be technically correct and complete, it is the business that must confirm that they are right in a business sense (that's why they're called 'business rules').

Validating is a joint responsibility of the business and IT people involved. Establish a working group early in the development of the data warehouse that has both business and technical people involved, and ensure that each rule has a 'business owner' who can resolve any doubts or differences when questions arise about a rule. Tip here: identify the owner when you first record the rule – it is much easier than trying to find a volunteer later when there is some issue to resolve!

As with data, validating metadata is an ongoing task. Expect your business rules workgroup to continue meeting occasionally after the data warehouse is in production, to review and update the rules.

METADATA AND CODE

Many business rules are eventually expressed as program code. This is especially true of transformation rules and calculation of derived data items, but also for extraction (especially if changed data capture is in use to read only the newest data) and for summarization.

Try to record these business rules so that they can be used to generate the necessary code. You can also programmatically generate the published metadata pages from the same records. In this way you can be sure that the process and the metadata description of it are consistent. However, don't just publish code – most business rules start with a narrative text version from the business, so remember to provide this as well.

SAS/Warehouse Administrator® software is a good framework for this – it has built-in code generation, and an API that lets you read the same metadata to export in any format you like (Figure 9).
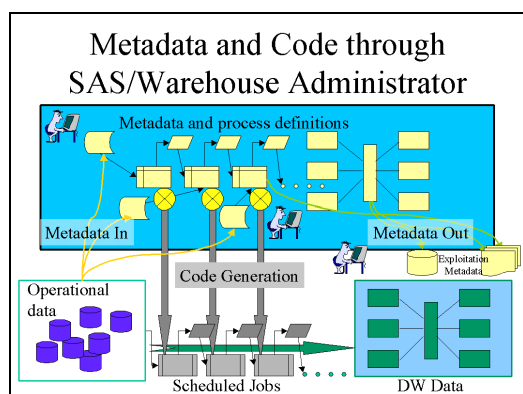

Figure 9

The use of fully connected metadata with the API supports code generation, automated publication of process rules and impact analysis from the same metadata. Impact analysis is an

important tool for maintenance and change in the data warehouse (Figure 10)
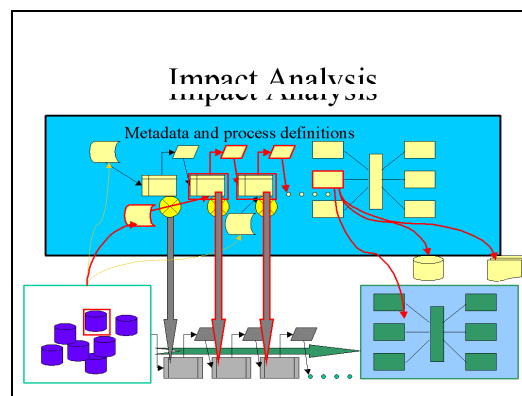

Figure 10

Making extra use of your metadata in this way helps repay the investment in collecting and organizing the metadata in the first instance - another form of ROI that is important to the value of the data warehouse

DATA QUALITY

Data quality is much discussed in data warehousing – and often organizations look to the data warehouse to improve data quality. This is a noble objective, but a considerable challenge!

Data will never be more accurate than at the moment they are collected and recorded from the real world. Certainly you may apply 'cleansing' rules later to make it more <u>consistent</u> – for example always spelling names or street addresses the same way – but this does not necessarily make the data more <u>accurate</u>!

So when you alter data for the data warehouse, make sure a report of what has been changed goes to the original owners of the data. Don't try to use a high correction rate as a stick to beat them with (after all they may not be able to improve their systems right away) but you do need confirmation from them that the changes to any value are correct and justifiable.

Feedback like this will not fix data problems overnight, but in the long term it should have the desired effect.

In the meantime, ensure that the error or correction rate is recorded in your metadata – and published so that data users have access to that information too.

LIVING WITH THE DATA WAREHOUSE

As well as the basic routine running, remember to monitor who is using the data warehouse and whether it continues to meet their needs. There should be an ongoing review process, where the original project steering committee becomes a 'management board' for the data warehouse. This is another reason why the sponsor must be committed for the long term.

Consider the life cycle of change in the data warehouse (more of those Seven League steps again!).

If the data warehouse is successful, there should be continuing need for change – new business requirements, new data sources to add, new subjects within the data store, new marts and applications for existing data.

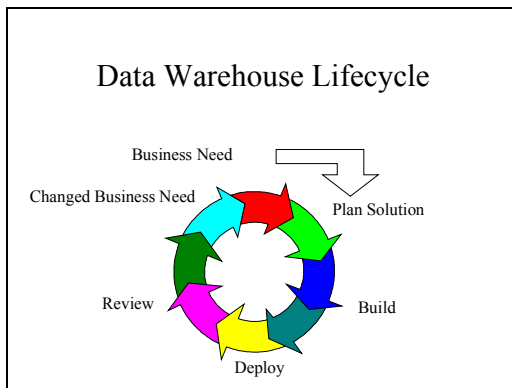This life cycle is continuous (Figure 11).

Data Warehouse Lifecycle

Figure 11

Engineering the data warehouse to support change is a considerable skill. It is unlikely you will be totally satisfied with your first effort – another reason to build in stages rather than all at once! Each time you add new data you will face the choice of extending an existing data structure or scrapping it to replace with a new one.

Small data marts might be considered cheap throwaway items – although we always want to get good value out of each one before discarding it, the disruption caused by replacing a single mart is usually small. But anything larger must be able to adapt and evolve.

For this reason we value flexibility over most other factors in a design for the detailed data stores of a data warehouse.

To support this we aim to get the fundamental dimensions for data correct early in the development of the data warehouse. Adding new descriptive attributes or fact measures to the data warehouse is comparatively easy – as are new ways of viewing dimensions with different hierarchies and so on – but new fundamental dimensions will involve major rework.

A FEW GOLDEN RULES

Everyone has their favorite do's and don'ts; here are some of mine.

SOME DON'T'S

Don't just solve today's business requirement alone. A narrow point solution will not be adaptable to changing business needs, and can only be thrown away and replaced when the time comes to change. This can be a difficult concept to 'sell' to your sponsor if all they want is a single solution – but they will thank you for it later when their next requirement comes along!

Don't expect to build, deploy and then disband the team. Running a data warehouse is not like maintaining and running a simple application – you will value the continuity of a core team throughout the life of the data warehouse. This does not mean you can't augment with extra people when times are busy, but do maintain the core team for the long term.

Don't assume everyone knows what the data mean. Even when everyone says they do understand, experience says that this is usually not true. Be prepared to re-state and validate everything, especially business rules – and make sure these are reviewed by more than one department, especially the financial rules.

SOME DO'S

Do plan for the lifetime of the data warehouse. An important part of this will be the review process. Form your working group of business and technical people to resolve questions and issues with the data, business rules and definitions. Expect to add to this group over time, or to alter responsibilities there, but do continue to meet occasionally as part of the ongoing review.

To support this, aim for long term sponsorship – you cannot keep a data warehouse growing and changing without a supporting sponsor. If your sponsor is moving on, make it a priority for them to 'sell on' the sponsorship higher within the organization before they go. If you wait until they are no longer there it will be too late, and you will have extra work to find new sponsorship.

Do expect the data warehouse to change over time. It has been said that a data warehouse that does not change is a dead data warehouse. Since change is inevitable, be ready for it by monitoring how business needs are being met and be on the lookout for new requirements.

Do check quality early and often. Late discovery of flaws in data or business rules is a major cause of pain in data warehouse projects. Always read some data whilst compiling business rules to help you validate the rules – data will often be less consistent than the first version of your rules will expect. Finding this early will allow time to improve the rules.

Aim for a preliminary load of data at the earliest possible point in the project and have it checked. If you have passed the halfway point in elapsed time for construction and test without doing this, then you are starting to run out of time to identify and correct the inevitable imperfections. Successful preliminary load is an important milestone!

Do break your data warehouse down into manageable steps. Incremental building is lower-risk and gives earlier return on investment. If you really can't break it down into deliverable smaller steps, then see the next point.

Do divide large single builds with definite milestones. Avoid those "Seven League" stretches by making sure you "touch ground" at key points along the way.

CONCLUSION

Building and running a data warehouse requires commitment. This is true whether your are allowed to call your project 'data warehouse' or not.

It is a long journey, with many steps – and although some of those steps are large, with some care you can avoid excessive strain. Be on the lookout for over-large steps, and define intermediate milestones. Make each milestone something you can deliver and your likelihood of success will be high.

REFERENCES

W H Inmon ('Bill Inmon') is the author of many books and papers on data warehousing, and is credited with being the 'father of data warehousing'. He is an advocate of incremental building of the enterprise data warehouse based upon defined data models. For details see his website at www.billinmon.com

Dr Ralph Kimball is the author of many books and papers on data warehousing, and is credited with inventing the 'star schema' for dimensional data warehousing. He advocates building incrementally based upon a 'bus architecture' of connected dimensional structures. For details see his website at www.ralphkimball.com

"Getting Started and Finishing Well" by Peter Nolan, edited by Ralph Kimball, published in the Data Webhouse column of *Intelligent Enterprise* magazine, May 7th 2001. Full article available at
www.intelligententerprise.com/010507/webhouse1_1.shtml

Figures 1 through 8 are reprinted from *Warehouse Architecture Course Notes, Edition 2.1.* Copyright(c) 2001, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective owners.

CONTACT INFORMATION

SHORT BIOGRAPHY OF THE AUTHOR

When I first encountered SAS in 1981, I already had over seven years IT experience in management information, systems management and application development. SAS was a revelation to me in power and flexibility for data management and analysis. Since that time, all my work has involved SAS in some way!

Joining the then-new European SAS Institute operation, I managed and grew the UK technical support and customer services over several years. Along the way, I performed almost every technical role, from customer support and education, through application system design and development, to management of the technical division.

From the early 1990's I have concentrated on the practical application of SAS technologies for data warehouse and related systems, with an emphasis on efficient processing and the best use of tools and metadata for low-maintenance systems. This has involved planning, architecture and design for several projects, and technical review and advice for many more.

Since 1997 I have worked as an independent consultant, specializing in SAS-based data warehouse and data mart architecture planning, design and performance improvement, and the use of software tools in data warehousing.

The past year and a half has seen me writing and presenting data warehouse architecture, design and methodology training for SAS in Europe, whilst also supporting projects through assessment, review and planning work.

I continue to find solving the challenges of data warehousing a stimulating experience, and enjoy applying my skills to these to meet a wide range of business needs.

CONTACT DETAILS

Your comments and questions are valued and encouraged.

Contact the author:-

Steve Morton
Applied System Knowledge Ltd.
51 Blandy Road
Henley-on-Thames
RG9 1QB
England
Work Phone: +44 1491 411977
Email: steve_morton@ask.u-net.com
Web: www.ask.u-net.com