Data Warehousing Concepts

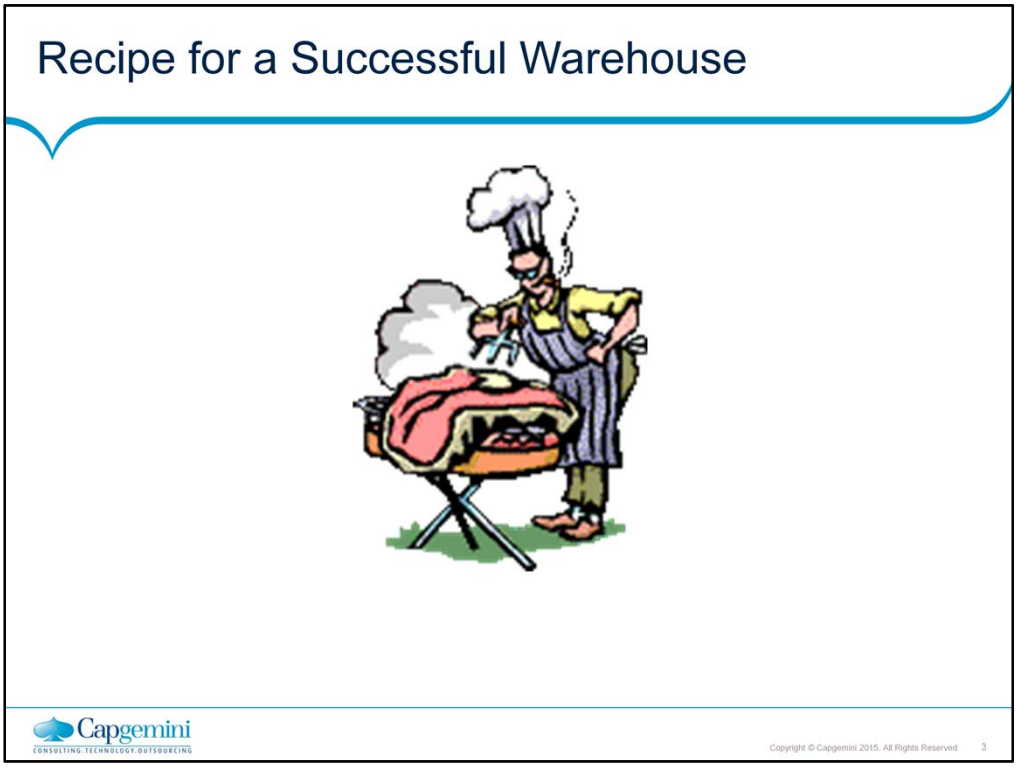
Lesson 7: Best Practices for Building Data Warehouse

Lesson Objectives

- In this lesson, you will learn:
  - Requirement for successful Data Warehouse
  - Data warehouse pitfalls
  - Popular BI DW tools and suits
  - Trends in BIDW

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2



From day one establish that warehousing is a joint user/builder project
Most of the Warehouse projects will fail if the builders get specs from the users, go off for 6 months, and then come back with the 'finished' project. Warehouses are iterative! (Hear I put the word iterative means there are lots of mistakes in the projects.) Builders and users working with each other will not reduce the number of iterations, but it will reduce the size of them.

Establish that maintaining data quality will be an ONGOING joint user/builder responsibility
Organizations undertaking warehousing efforts almost continually discover data problems. Best to establish right up front that this project is going to require some additional ongoing responsibility.

Train the users one step at a time
Typically users are trained once. In several days they learn both the basics and intermediate and sometimes advanced aspects of using a tool. Slow down! Consider providing training initially in the minimum needed for the user to get something useful from the tool. Then let the user use the tool for a while (meaning several days, weeks, or months). Having basic training and some hands on experience, the user will have a much better context with which to grasp the next level. Also, once the basics and the next level are learned, keep training the users! After a year using the tool, schedule advanced training.

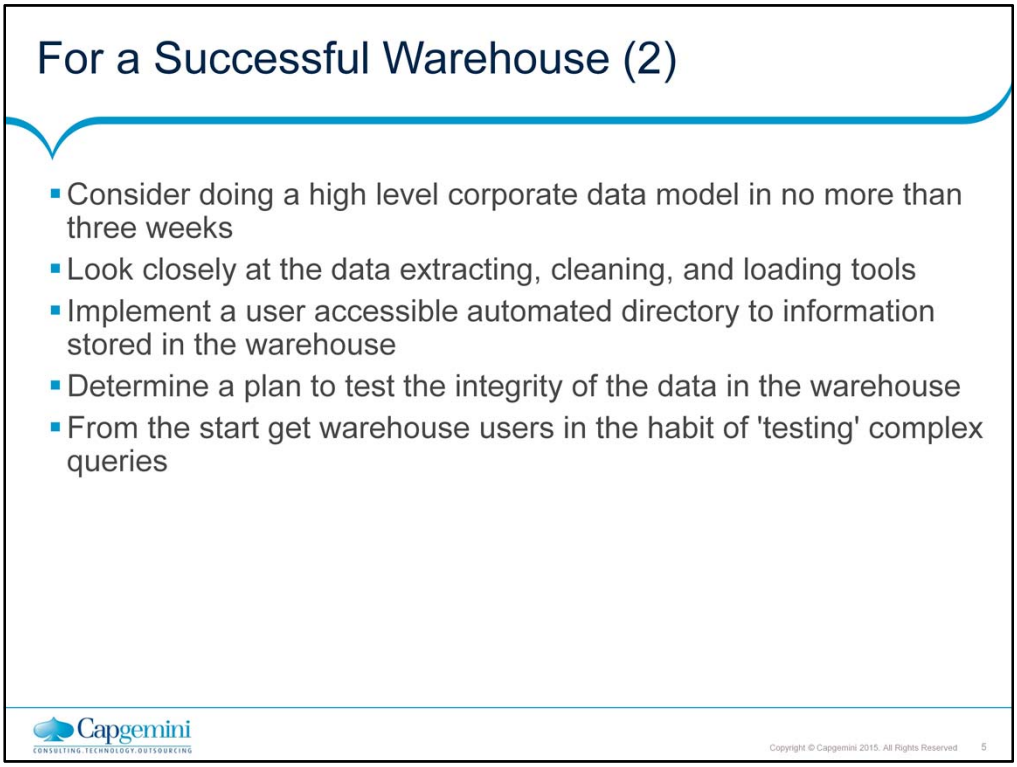
Train the users about the data stored in the data warehouse
Users often need more training about the stored data than about the tools used to access the data. One should not assume the data are self-explanatory or that any metadata you may provide will answer any questions. Note that users are often used to seeing data in canned reports and seeing data in its "raw" form can be confusing.

For a Successful Warehouse (1)

- From day one establish that warehousing is a joint user/builder project
- Establish that maintaining data quality will be an ONGOING joint user/builder responsibility
- Train the users one step at a time
- Train the users about the data stored in the data warehouse

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 4



Consider doing a high level corporate data model / data warehouse architecture "exercise" in three weeks
Actually, the key point regarding time is to "time-box" the exercise into a relatively short time. After about three weeks, the marginal benefits from additional time devoted to these types of exercises rapidly decrease. - The corporate model is going to identify, at a high level, subjects and relationships and most importantly, what are the chunks of information that it makes sense to deliver in different projects. The architecture part of the exercise to determine the dimensions, definitions of derived data, attribute names, and information sources that you will attempt to use consistently in your data warehousing efforts. The exercise also consists of coming to an agreement as to how to keep the corporate model up-to-date and how to make sure future data warehousing efforts pay attention to the architectural principles.

Implement a user accessible automated directory to information stored in the warehouse

The majority of successful warehousing efforts I have seen included providing some means for the warehouse user to locate stored information. Most of the times this involved building a separate database with directory information. And most of the time, a pretty simple database sufficed for initial use.

Once you know what raw data you want to feed into the data, request that data

If you have done some reading on data warehouse development you probably have read that figuring out the process of extracting, transforming, and loading (ETL) usually takes the majority of the time in initial data warehouse development. In project management lingo, figuring out ETL is usually on the critical path. - If you know what raw data you need, request it as soon as you know it. You are probably going to have to ask one of the programmers of the legacy feeder systems to initially get this data for you.

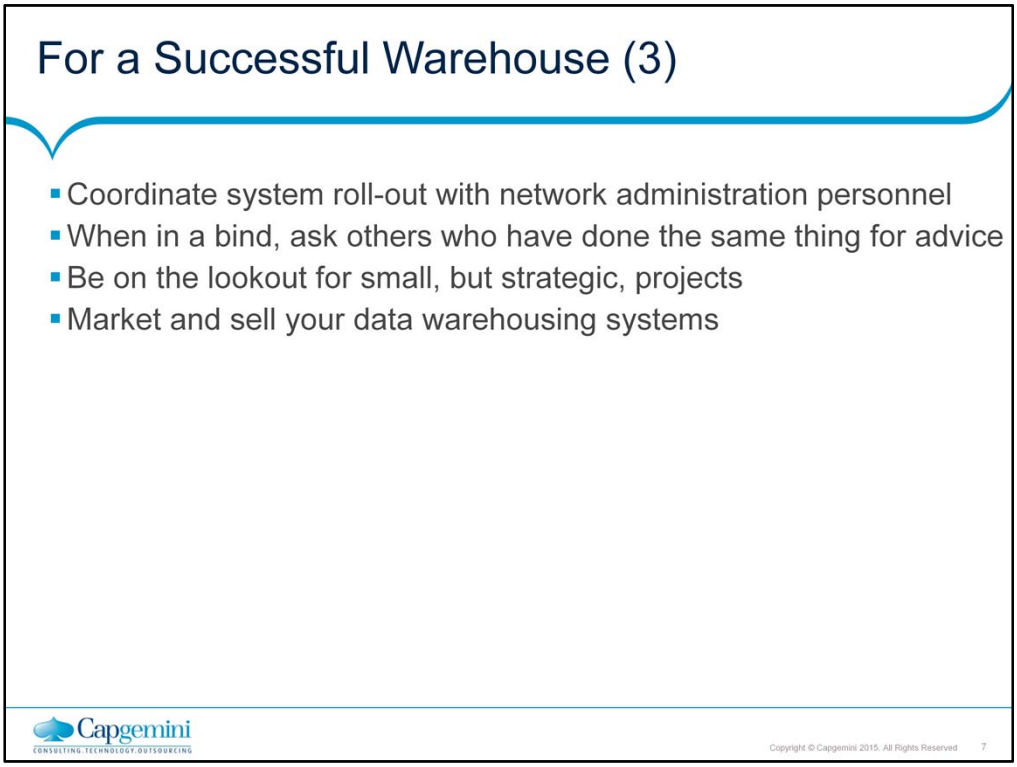
For reasons of politics, overwork, and just plain lack of knowledge of how data are physically stored in a system, the feeder system programmer often can take a while to get you that data.

Determine a plan to test the integrity of the data in the warehouse

Do not underestimate the importance of user faith in the integrity of the warehouse data. Huge warehouse efforts quickly go sour if after system roll-out users find multiple mistakes. A good investment of time in the initial stages of a warehouse project is for the builder and user to jointly determine what checks will be made on the warehouse data during development and what checks need to be made on an ongoing basis. The checks including tying warehouse data controls back to controls in feeder systems, checking the correctness of aggregation logic, testing whether classifications codes were assigned correctly.

From the start get warehouse users in the habit of 'testing' complex queries

Many people will assume that the query result is correct. At the very least, get the user in the habit of eyeballing the query or report to check if several records that should be included are, in fact, included and that several records that should not be included are, in fact, not included.



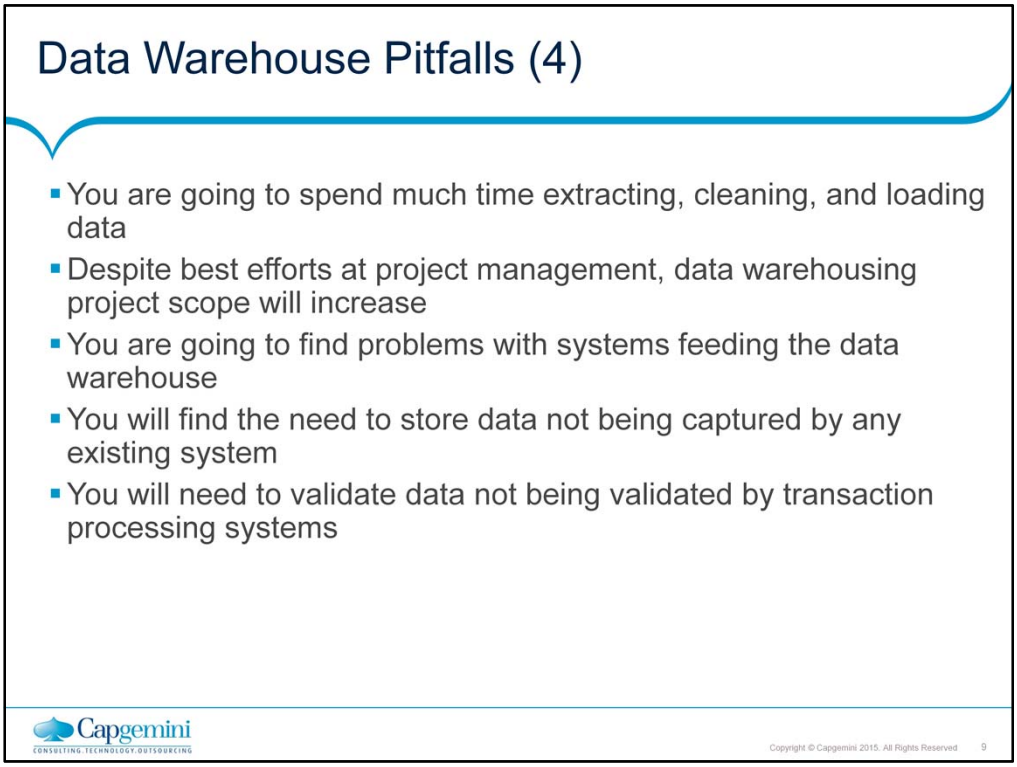
Coordinate system roll-out with network administration personnel
Use of data warehousing systems can bring about some strange spikes in network activity. If you keep network administration people informed of the roll-out schedule, chances are they will monitor network activity for you and be ready to make adjustments to the network as necessary.

Have a good grasp of desktop databases and spreadsheets
Even if you are dealing with a 100 TB database, there are so many little tasks to be done in a data warehousing project where knowledge of these tools will be helpful. Skillful use of these tools during development can be a huge productivity enhancer.

Be prepared to support beginning users immediately and at any time
We developers often greatly underestimate users' hesitation to begin using the data warehouse. This hesitation could be because of user fear of technology or user fear that they will not get Information System support. So, the first point is to be available to help when the user wants to try to use the data warehouse the first time. Users also may want to use the data warehouse for the first time during the weekend or at 6:00 in the morning or 8:00 at night. The distractions are less at those times. If you want to make that beginning user as a committed customer of your data warehouse, you better be available to support the user when he starts out whatever the day or the hour.

Maintain the audit trail to the feeder systems
That is, make it as easy as possible to tie the data in the data warehouse to the feeder systems. Your users have to trust the numbers in the data warehouse. You owe this to the users in order to maintain their trust.

Market and sell your data warehousing systems
For the most part, use of data warehousing systems is optional. This means you have to identify the potential users of the systems, help them understand what are the benefits of the system, and then make them want to keep coming back to use the system.



You are going to spend much time extracting, cleaning, and loading data
The usual figure quoted is that approximately 80% of the time building a data warehouse will be spent on this type of work. (No one has ever explained how this percentage was obtained though.) Suffice it to say, though, the amount of time on these tasks is often grossly underestimated. Note that this point is about extracting and cleaning and loading. Though by now many people are aware the cleaning the data is complex, extracting data and loading data are equally, if not more, complex.

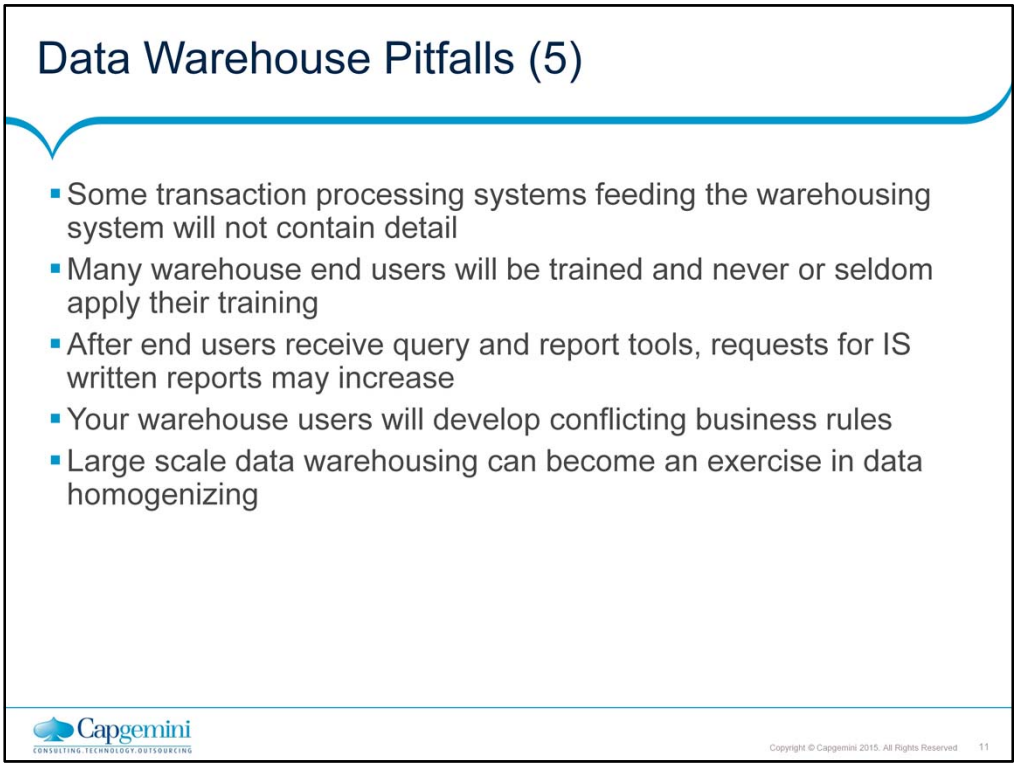
Despite best efforts at project management, data warehousing project scope will increase
To paraphrase data warehousing author W. H. Inmon, traditional projects start with requirements and end with data. Data warehousing projects start with data and end with requirements. Once warehouse users see what they can do with 2000's technology, they will want much more. (Which is fine!)

One piece of advice for the warehouse builder is never to ask the warehouse user what information he wants. Rather, ask what information he wants next.

You are going to find problems with systems feeding the data warehouse
Problems that have gone undetected for years will pop up. You are going to have to make a decision on whether to fix the problem in what you thought was the 'read-only' data warehouse or fix the transaction processing system

You will find the need to store data not being captured by any existing system
A very common problem is to find the need to store data that are not kept in any transaction processing system. For example, when building sales reporting data warehouses, there is often a need to include information on off-invoice adjustments not recorded in an order entry system. In this case the data warehouse developer faces the possibility of modifying the transaction processing system or building a system dedicated to capturing the missing information.

You will need to validate data not being validated by transaction processing systems
Typically once data are in warehouse many inconsistencies are found with fields containing 'descriptive' information. For example, many times no controls are put on customer names. Therefore, you could have 'DEC', 'Digital' and, 'Digital Equipment' in your database. This is going to cause problems for a warehouse user who expects to perform an ad hoc query selecting on customer name. The warehouse developer, again, may have to modify the transaction processing systems or develop (or buy) some data scrubbing technology.



Some transaction processing systems feeding the warehousing system will not contain detail
This problem is often encountered in customer or product oriented warehousing systems. Often it is found that a system which contains information that the designer would like to feed into the warehousing system does not contain information down to the product or customer level. By the way, this is what some people label a 'granularity' problem.

You will under budget for the resources skilled in the feeder system platforms
In addition to understanding the feeder system data, you may find it advantageous to build some of the "cleaning" logic on the feeder system platform if that platform is a mainframe. Often cleaning involves a great deal of sort/merging - tasks at which mainframe utilities often excel. Also, you may find that you want to build aggregates on the mainframe because aggregation also involves substantial sorting.

Many warehouse end users will be trained and never or seldom apply their training
I once read a study that claimed that only one quarter of the people who get training in a query tool actually become heavy users of the tool.

After end users receive query and report tools, requests for IS written reports may increase
This phenomenon was seen with many of the information centers of the 1980s. It comes about because the query and report tools allow the user the users to gain a much better appreciation of what technology could do. However, for many reasons the users are unable to use the new tools themselves to realize the potential. By the way, if this happens do some honest research on why. Granted there are many reports that are so complex that Information Systems expertise is going to be required no matter what tool the end user has. However, many times this phenomenon points to training needs.

Your warehouse users will develop conflicting business rules
Many warehouse tools allow users to perform calculations. The tools will allow users to perform the same calculation differently. For instance, suppose you are summarizing beverage sales by flavor category. Also suppose that the flavor category includes cherry and cola. If you have a cherry cola brand there is a chance that two users will classify the brand in different categories. You will find that there are means to incorporate some of the business rules in your warehouse. However, the number of possible business rules is so large that you will not be able to incorporate all rules

Your warehouse users may not know how to use data
After many years of using whatever reports have been thrown in their faces, the users may not know what data to use their newfangled decision support tools to retrieve. To use a phrase from pop sociology, the users have been "culturally conditioned" to use what they are given and to never ask for more.

Large scale data warehousing can become an exercise in data homogenizing
Data have quirks! Sometimes when we developers combine detailed data for different subjects, in our efforts to make everything 'fit' we can take the life out of the data. For instance, if your company sells dog food and auto tires, you want to be careful if you are building a sales data warehouse for both lines of business. You have to make a judgment call as to whether these businesses fit the same logical and/or physical model.

Data Warehouse Pitfalls (6)

- 'Overhead' can eat up great amounts of disk space
- The time it takes to load the warehouse will expand to the amount of the time in the available window... and then some
- Assigning security cannot be done with a transaction processing system mindset
- You are building a HIGH maintenance system
- You will fail if you concentrate on resource optimization to the neglect of project, data, and customer management issues and an understanding of what adds value to the customer

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 13

'Overhead' can eat up great amounts of disk space
A popular way to design a decision support relational databases is with star or snowflake schemas. Persons taking this approach usually also build aggregate fact tables. If there are many dimensions to the data, be aware that the combination of the aggregate tables and indexes to the fact tables and aggregate fact tables can eat up many times more space than the raw data. If you are using multidimensional databases, be aware that certain products pre-calculate and store summarized data. As with star/snowflake schemas, storage of this calculated data can eat up far more storage than the raw data.

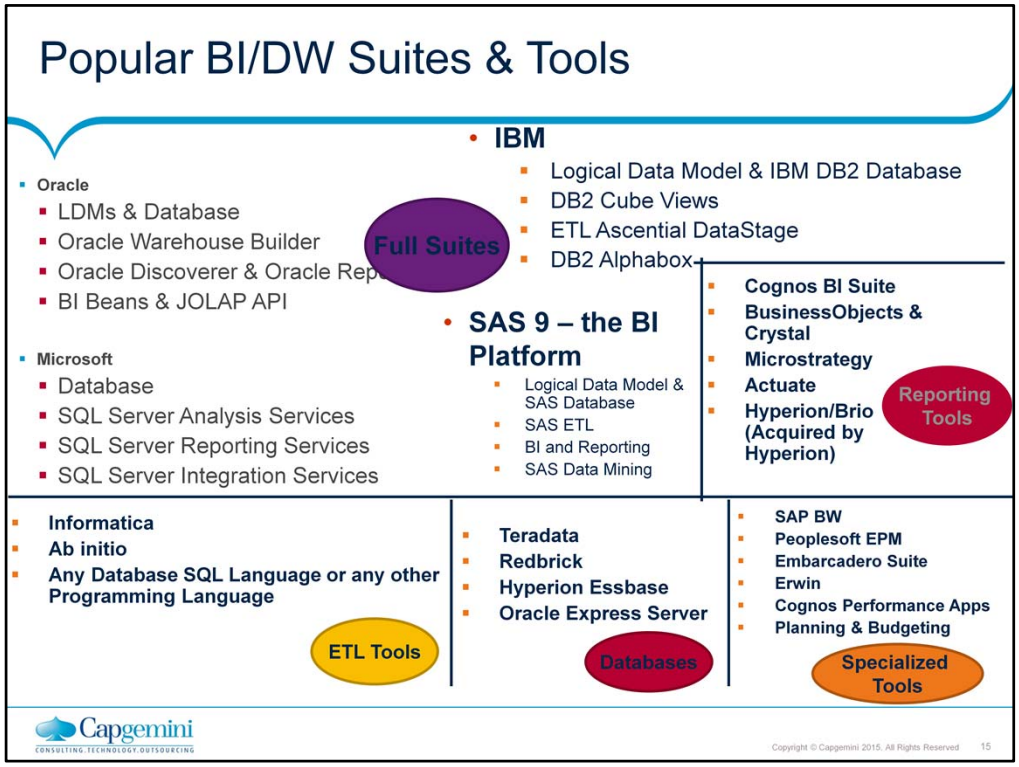
The time it takes to load the warehouse will expand to the amount of the time in the available window... and then some
You will do yourself well by understanding the different ways to approach updating the warehouse. Before you decide that you can do complete refreshes, be aware that "There's all day Sunday to load the database!" have been famous last words of more than a handful of warehouse developers.

You are going to have a tough problem with security - especially if you make your data warehouse Web-accessible
You are going to face a paradox - the more accessible you make your data warehouse (and by accessible, I don't just mean making it Web accessible - I mean architecting it in a way that people want to use it), the greater security risk you are exposing yourself too. Frankly, restricting people to "need to know" does not cut it in the organization on the 2000s. But, on the other hand, exposing information to theft from anyplace in the globe is not too great for job security either.

The data warehouse data you do not reconcile with the feeder systems will cause the problems
For certain data warehouse data you are going to think that there is no logical way that data in the feeder systems can be reconciled with what are in the warehouse. Then, when a user looks at a report and tells you "I think there is a problem", it will be with the unreconciled data. Unfortunately, you will then discover there is a way, albeit roundabout, to reconcile the data.

You are building a HIGH maintenance system
Reorganizations, product introductions, new pricing schemes, new customers, changes in production systems, etc. are going to affect the warehouse. If the warehouse is going to stay 'current' (and being current will be a big selling point of the warehouse), changes to the warehouse have to be made fast.

You will fail if you concentrate on resource optimization to the neglect of project, data, and customer management issues and an understanding of what adds value to the customer
If you provide a system that is fast and technically elegant but adds little value or has suspect data, you will probably lose your customer from day one and will have a tough time getting him back. For the most part, use of data warehousing systems is optional. The customer has to want to use the system.



There are lot of BI tools in the market. The Organization like Oracle, Microsoft , IBM and SAS providing tools which provides end to end solutions that includes Designing , Profiling, MetaData, ETL , Database and Reporting solutions.

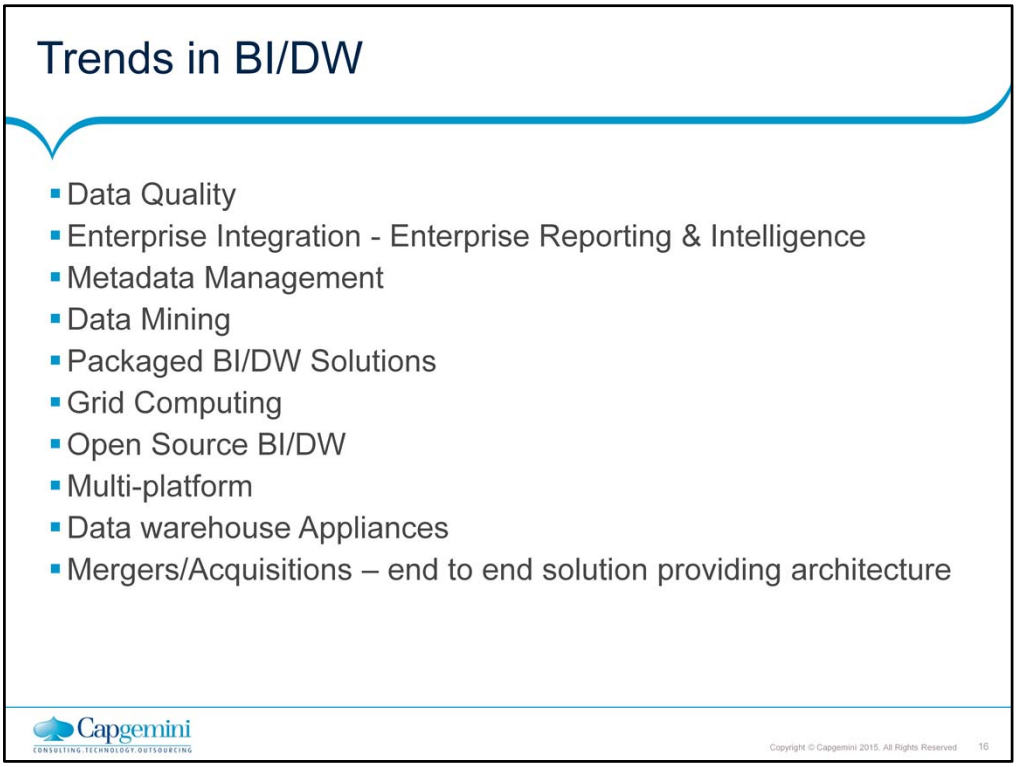
There are exclusive ETL tools such as Informatica, DataStage, Business Object Data Integrator, OWB , Abinitio which provides Extraction , Transformation and Loading solutions and handles huge volumes of data.

There are exclusive Database tools like Teradata, Redbrick etc which provides database solutions to hold huge amount of data.

There are exclusive Reporting tools like cognos, Business Objects XI, Actuate etc which provides Reporting solutions for various users view. And also comfortable with drill down, roll up, drill across, slice, dice operations.

In addition, there are some specilized tools which can used for specific purpose for instance Erwin would be used for designing database etc.

In every tools lot of enhancements are taken place and most tools supports for SOA (Service Oriented Architecture), Data Integration , Data Quality and Cloude Computing.



In this presentation we can discuss about emerging Trends in Business Intelligence and Data Warehouse. Following are the Emerging Area where BIDW is playing vital role.
Data Quality – As Industry needs quality data to take decision hence most of the tools are providing solution to provide quality data for instances in DataStage Quality stage has embaded in DataStage 8x, In BOBJ Data Services they provides transformers which supports for Quality.

Most of the tools supports and maintain **Metadata** and keep track of every metadata it maintains. Tools provides to generate impact analysis report for metadata.

Data Mining is equally playing vital role in industry like Insurance, Telecommunication, Banking etc. There are lot of tools uses different algorithms.

Many organization provides end to end solutions including software and **hardware appliances**.

Every tools are enhancing new features to work with **multiple platforms**, multiple database and multiple architecture.

As data is growing in million billions of records as a result performing complex operations through single computing system may not be sufficient hence most of the tools are providing **Grid computing facility** in where Data can be routed across multiple computing systems to perform complex tasks.

Open Source BIDW are also emerging trends and so that one customize the tool based on their requirements.

Summary

- In this lesson, you have learnt:
  - Precautions to be taken for successful data warehouse
  - Tools available for data warehouse
  - Data warehouse trends

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 17