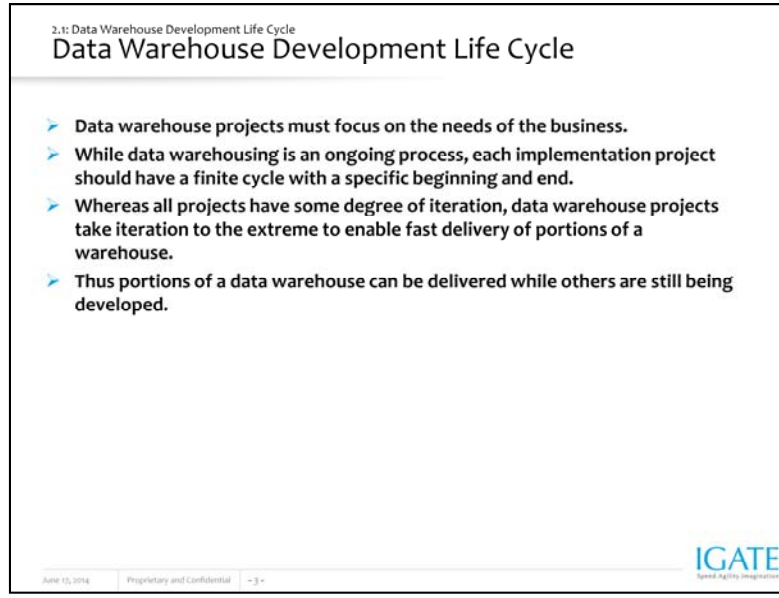

BI Software Development Lifecycle
Lesson 2: BI Project Execution Methodology (Development)
June 13, 2014
Proprietary and Confidential
- 1 -
IGATE

Lesson Objectives

- **List the coverage for this lesson**
  - Data Warehouse Development Life Cycle
    - Manage the Project
    - Define the Project
    - Analysis
    - Design
    - Construction
    - Deploy
    - Maintenance
  - Warehouse Model throughout the Life Cycle
  - Development Methodology (IIDM)



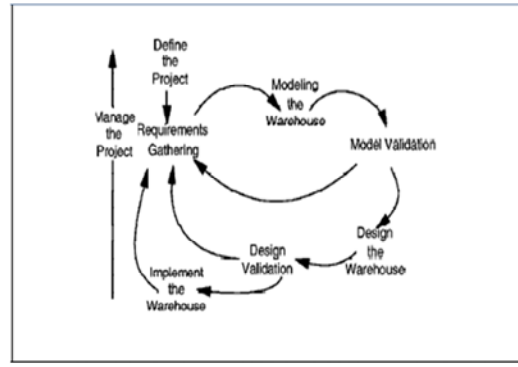
Any Business Intelligence development methodology should take into account at least the following very important aspects:

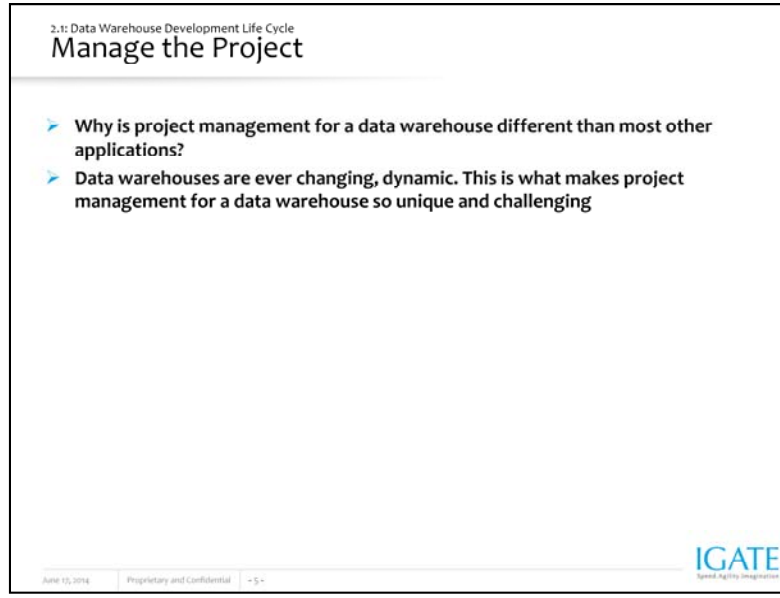
• Business Intelligence development should be business driven at all times. Alignment between business and IT is crucial and should be established at the early beginning of the program and managed continuously during the whole program.

• Business Intelligence development is a continuous improvement process. Architecture and infrastructure should be selected and implemented with the future in mind. Development on the other hand must be carried out in short iterations delivering business value rapidly. A BI life cycle management process is required to support continuous improvement.

2.1: Data Warehouse Development Life Cycle

Data Warehouse Development Life Cycle (Contd…)

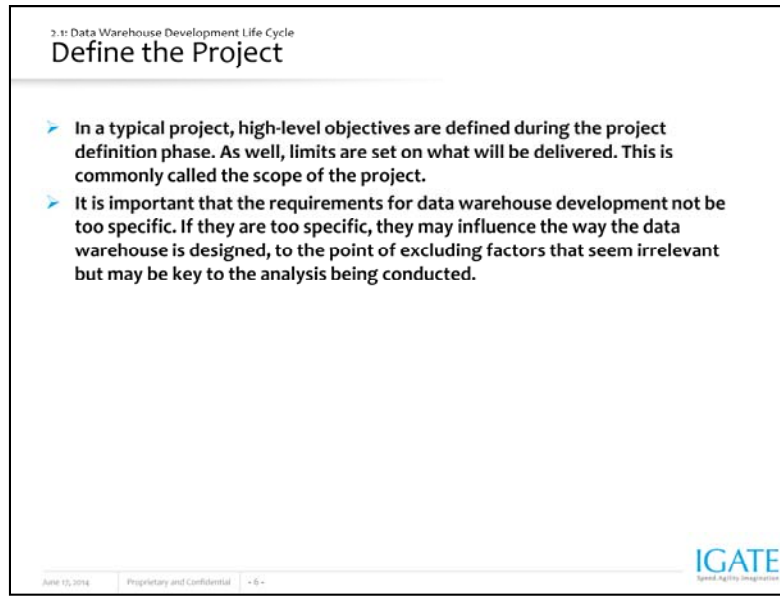




The difference is between managing the project and managing the data warehouse is that management of a project is finite in scope and is concerned with the building of the data warehouse, whereas management of a data warehouse is ongoing (just as management of any other aspect of your organization, such as inventory or facilities) and is concerned with the execution of the data warehousing processes.

There are two paths to project managing a data warehouse.

The first is to approach the project strictly from a project management perspective by managing the project scope and timeline. The second route is to follow through with the traditional responsibilities of project management and, at the same time, do a deep dive into the inner workings of the data warehouse.

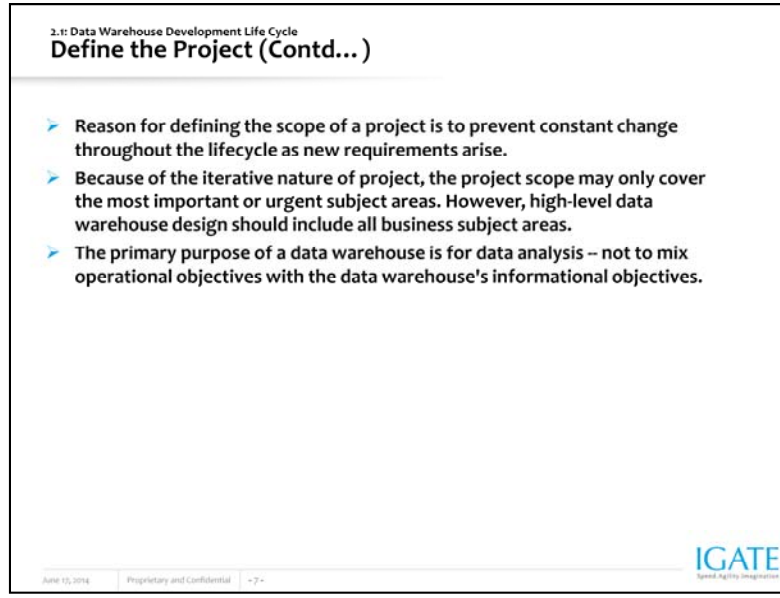


In a typical project, high-level objectives are defined during the project definition phase. As well, limits are set on what will be delivered. This is commonly called the scope of the project.

In data warehouse development, although the project objectives need to be specific, the data warehouse requirements are typically defined in general statements. They should answer such questions as, What do I want to analyze, and why do I want to analyze it? By answering the why question, we get an understanding of the requirements that must be addressed and begin to gain insight into the users information requirements.

It is important that the requirements for data warehouse development not be too specific. If they are too specific, they may influence the way the data warehouse is designed to the point of excluding factors that seem irrelevant but may be key to the analysis being conducted.

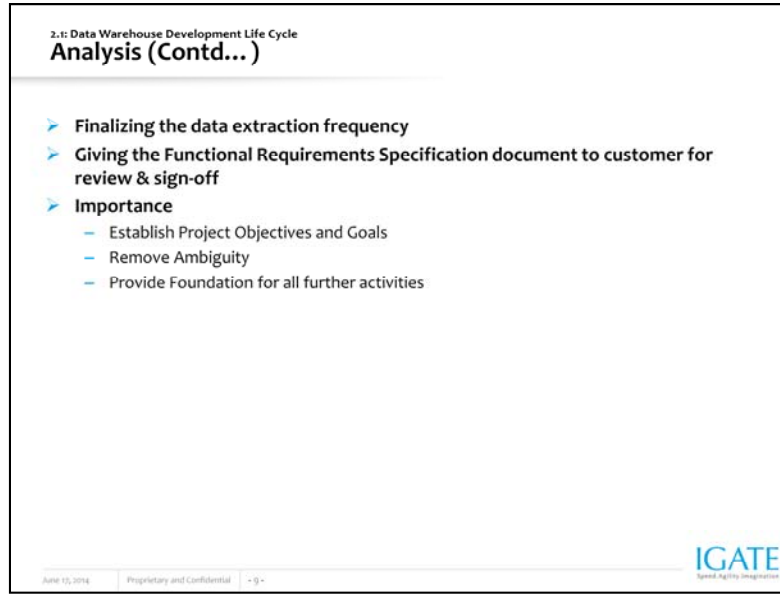
One of the main reasons for defining the scope of a project is to prevent constant change throughout the life cycle as new requirements arise. In data warehousing, defining the scope requires special care. It is still true that you want to prevent your target from constantly changing as new requirements arise. However, two of the keys to a valuable data warehouse are its flexibility and its ability to handle the as yet unknown query. Therefore, it is essential that the scope be defined to recognize that the delivered data warehouse will likely be somewhat broader than indicated by the initial requirements.

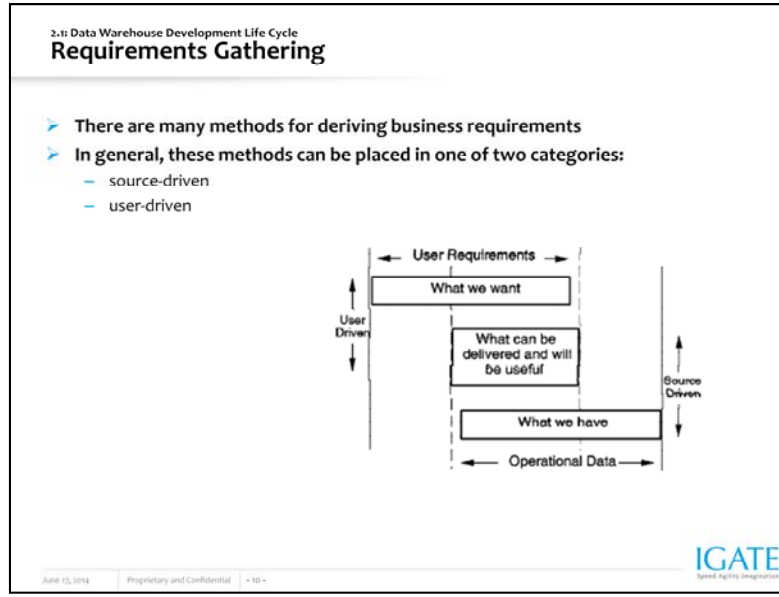

2.1: Data Warehouse Development Life Cycle
Define the Project (Contd...)
Reason for defining the scope of a project is to prevent constant change throughout the lifecycle as new requirements arise.
Because of the iterative nature of project, the project scope may only cover the most important or urgent subject areas. However, high-level data warehouse design should include all business subject areas.
The primary purpose of a data warehouse is for data analysis -- not to mix operational objectives with the data warehouse's informational objectives.
June 17, 2014
Proprietary and Confidential
- 7 -
IGATE

2.1: Data Warehouse Development Life Cycle

Analysis

- **The Analysis phase activities include:**
  - Identify the data sources
  - Identifying tables & columns, flat files and other source entities, which will provide data for the reports
  - Performing gap analysis of the data provided by source systems & required for developing the reports
  - Documenting all the calculations required for creating the reports
  - Giving recommendations for capturing missing data elements as a part of the system wherever applicable

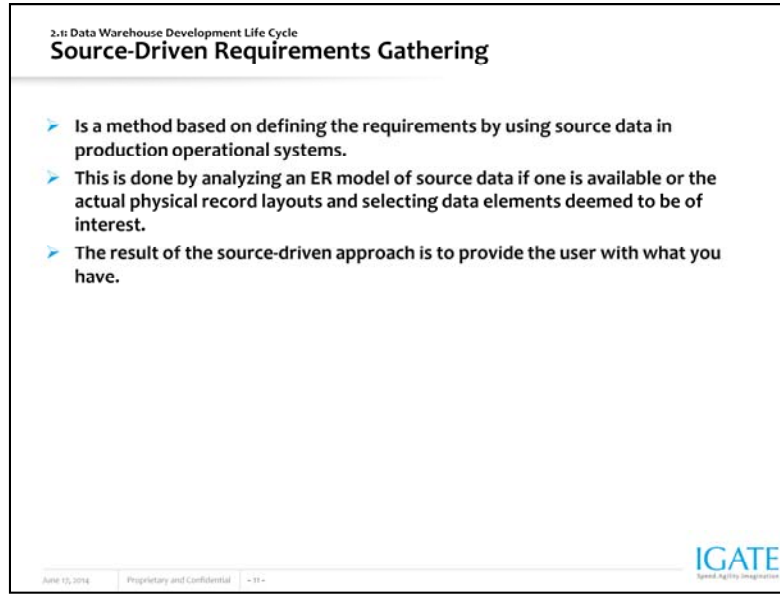

2.1: Data Warehouse Development Life Cycle
Analysis (Contd...)
Finalizing the data extraction frequency
Giving the Functional Requirements Specification document to customer for review & sign-off
Importance
Establish Project Objectives and Goals
Remove Ambiguity
Provide Foundation for all further activities
IGATE



The data warehouse development cycle focuses on facilitating the analysis that will change the process to make it more effective. Efficiency measures how much effort is required to meet a goal. Effectiveness measures how well a goal is being met against a set of expectations.

The requirements identified at this point in the development cycle are used to build the data warehouse model. But, the requirements of an organization change over time, and what is true one day is no longer valid the next. How then, do you know when you have successfully identified the users requirements? Although there is no definitive test, we propose that if your requirements address the following questions, you probably have enough information to begin modeling:

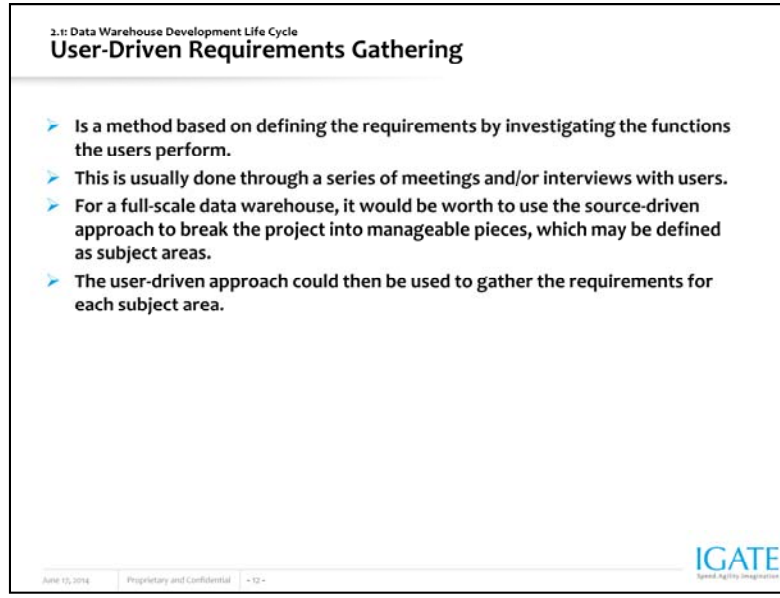
· Who (people, groups, organizations) is of interest to the user?
· What (functions) is the user trying to analyze?
· Why does the user need the data?
· When (for what point in time) does the data need to be recorded?
· Where (geographically, organizationally) do relevant processes occur?
· How do we measure the performance or state of the functions being analyzed?



The major advantage of this approach is that you know from the beginning that you can supply all the data because you are already limiting yourself to what is available. A second benefit is that you can minimize the time required by the users in the early stages of the project

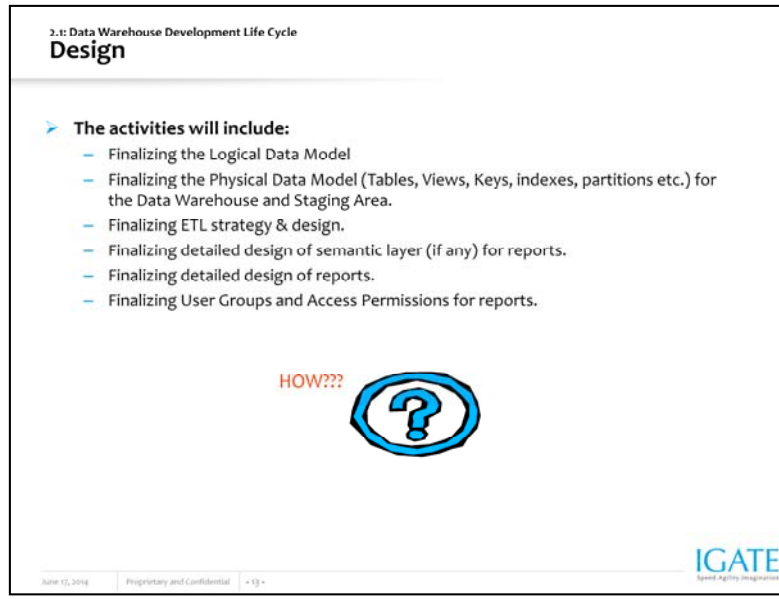
Disadvantage: By minimizing user involvement, you increase the risk of producing an incorrect set of requirements. Depending on the volume of source data you have, and the availability of ER models for it, this can also be a very time-consuming approach

The result of the source-driven approach is to provide the user with what you have. We believe there are at least two cases where this is appropriate. First, relative to dimensional modeling, it can be used to drive out a fairly comprehensive list of the major dimensions of interest to the organization. If you ultimately plan to have an organization wide data warehouse, this could minimize the proliferation of duplicate dimensions across separately developed data marts. Second, analyzing relationships in the source data can identify areas on which to focus your data warehouse development efforts.



The major advantage to this approach is that the focus is on providing what is needed, rather than what is available. In general, this approach has a smaller scope than the source-driven approach. Therefore, it generally produces a useful data warehouse in a shorter time span.

On the negative side, expectations must be closely managed. The users must clearly understand that it is possible that some of the data they need can simply not be made available. This is important because you do not want to limit what the user asks for. Outside-the-box thinking should be promoted when defining requirements for a data warehouse. This will prevent you from eliminating requirements simply because you think they might not be possible. If a user is too tightly focused, it is possible to miss useful data that is available in the production systems.

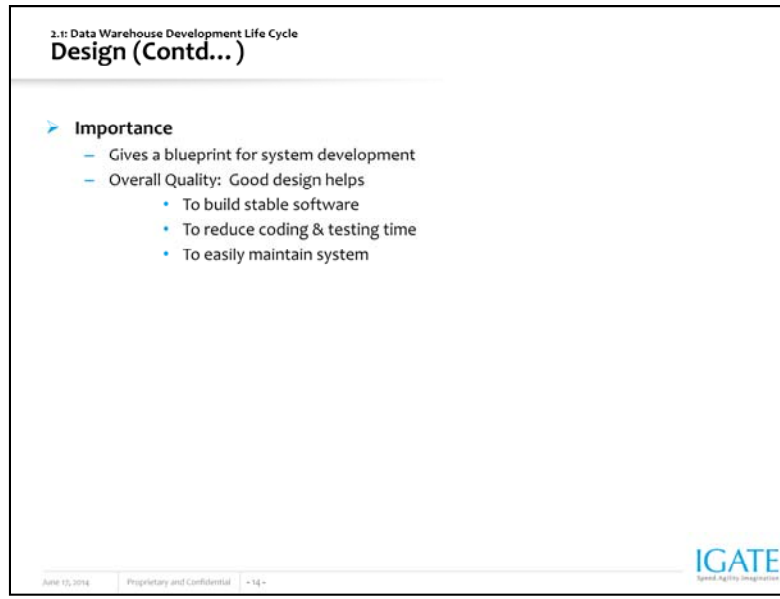


Solution design will follow the Analysis phase. Design will involve creation of the Technical Design Document based on the Functional Requirement Specification document. The activities will include:
Finalizing the Logical Data Model

- Identifying all the major entities and relationships and coming up with an Entity-Relationship diagram for the Data Warehouse
- Modeling the overall process that will support reporting needs.
- Creating a data dictionary

Finalizing the Physical Data Model (Tables, Views, Keys, indexes, partitions etc.) for the Data Warehouse and Staging Area.
Finalizing ETL strategy & design. The important design considerations will be:

- Finalize the data sources and file extracts
- Security
- Change Data Capture Mechanism for incremental loads
- Asynchronous vs. Synchronous Mode of Loading – concurrent processing of data streams
- Exception Handling & Management Strategy: Managing exceptions resulting from the business rule violations and re-processing them through the ETL (Extract, Transform and Load) processes, thus ensuring consistency in applying business rules.
- Detailed ETL specifications (Jobs, Source-to-Target mapping, scheduling etc.)



Finalizing detailed design of semantic layer (if any) for reports.
- Defining Name Spaces, Query Subjects & Query Items
- Defining/Reviewing relationships between query subjects
- Filters/Prompts to be included
- Calculations/aggregations to be included
- Packaging above objects

Finalizing detailed design of reports
- Report Functions and their definition
- Special considerations – Font, Color, Logo Considerations etc.
- Conditional Formatting Requirements
- Special functions/derived functions needed for analysis
- Business Functionalities of the report
- Scheduling, Refresh, Distribution, and Publish Requirements

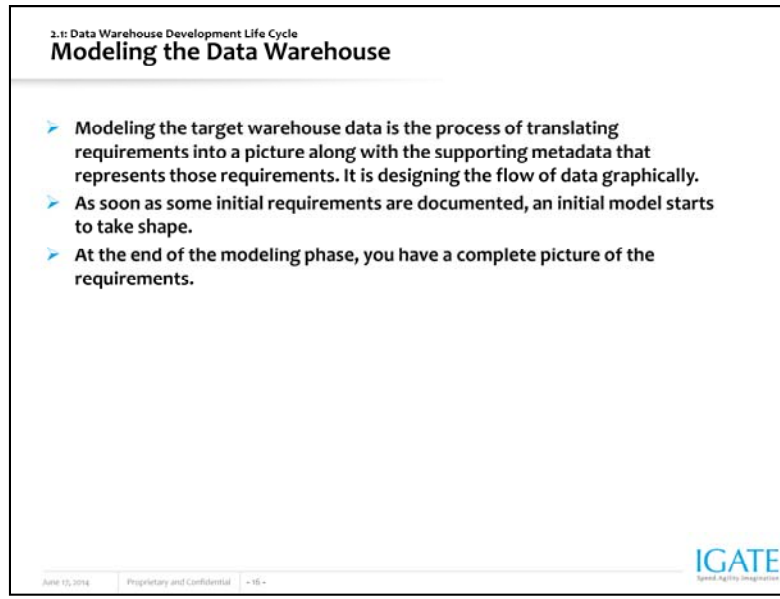
Finalizing User Groups and Access Permissions for reports
The technical specifications of ETL Transformations, semantic layer and reports will be collated into a Technical Design Document.
A comprehensive Test Plan and Test Cases will be created for System & Integration Testing as well as for Performance Testing.
Setup Development & Test Environment at offshore iGATE GDC.

Technical Design Document (containing Design and Specs for all the BI components, Logical and Physical design of the Data Warehouse) will be given to the customer for review and sign-off. Any further change in design should go through a change review and approval process involving both the iGATE point-of-contact and customer's IT/business users.

It is the design that gives the blueprint for software development. From here, the coding can begin.

The importance of design lies in the fact that it ultimately affects the success of the software construction, and its ease of maintenance. It provides representations of software, which can be assessed for quality. Without design, we risk building an unstable system - one that will fail when small changes are made or one that may be difficult to test.

It is observed that a good design helps in building a stable system; and reduces coding & testing time. It helps in building a robust system, and one that is easily adaptable to change.



Goal is to arrive at an understanding of the principal data sources and data elements of interest to the business or organization, and the relationships between the data sources, in order to satisfy requirements for information. There are two basic data modeling techniques: ER modeling and dimensional modeling

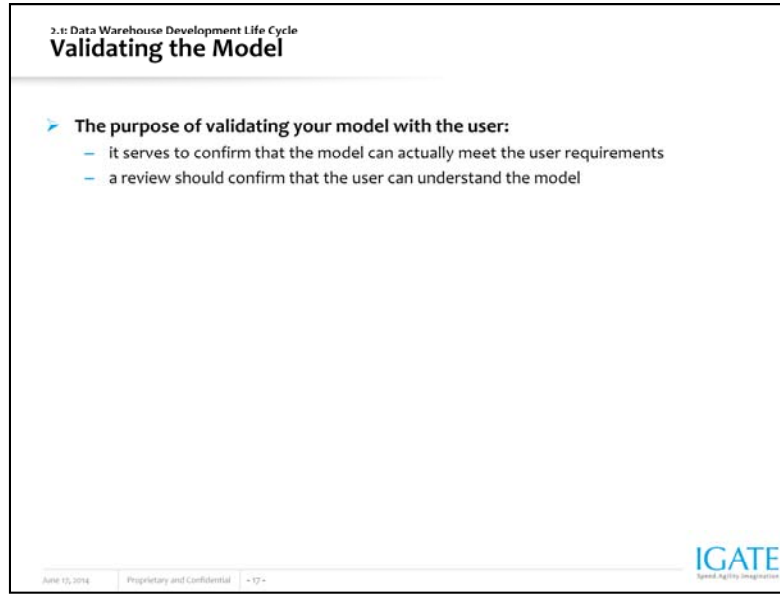
Creating an ER Model

ER modeling produces a data model of the specific area of interest, using two basic concepts: entities and the relationships between those entities. Detailed

ER models also contain attributes, which can be properties of either the entities or the relationships. The ER model is an abstraction tool because it can be used to understand and simplify the ambiguous data relationships in the business world and complex systems environments.

Creating a Dimensional Model

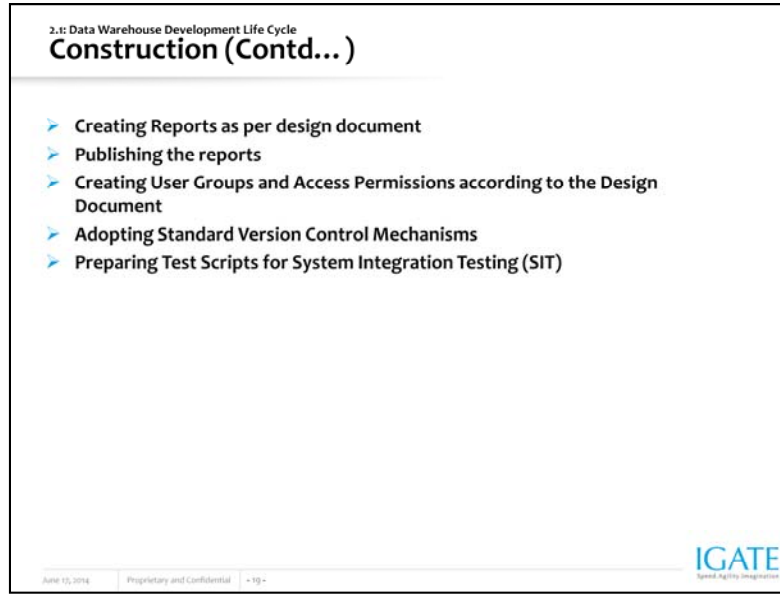
Dimensional modeling uses three basic concepts: measures, facts, and dimensions. Dimensional modeling is powerful in representing the requirements of the business user in the context of database tables.

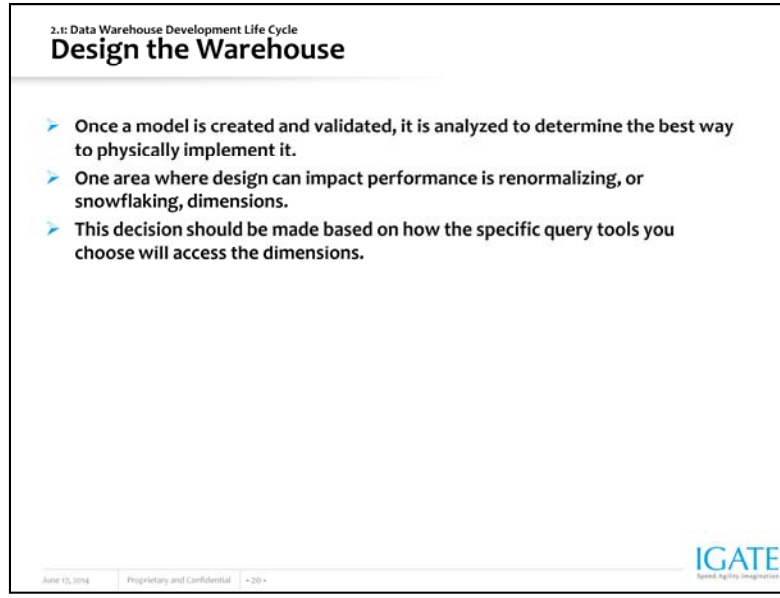


Validation at this point is done at a high level. This model is reviewed with the user to confirm that it is understandable. Together with the user, test the model by resolving how you will answer some of the questions identified in the requirements.
The iteration of development and the continued creation of partially complete models are the key elements that provide the ability to rapidly develop data warehouses.

The Build phase starts almost parallel to the Design phase with a lag of 1 week. Thus, the development of the Staging Area Data Model will start as soon as the Staging Area Data Model is designed, and the next Design activity continues in the meantime.
The Build phase will involve:

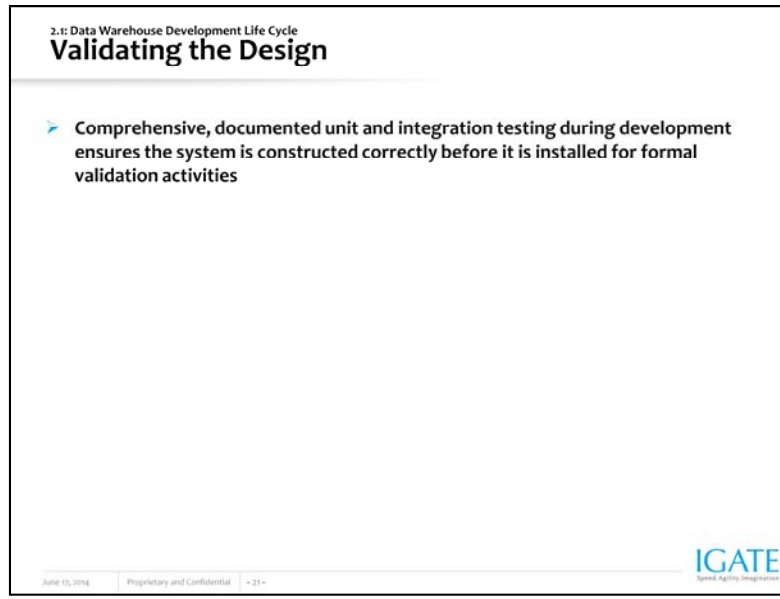
- Generating SQL scripts from Data Modeler (for Staging and Data-Warehouse)
- Creating the Staging Area and Data Warehouse using the Data Modeler scripts
- Developing ETL Transformations, Mappings and Workflow according to the Design Document and iGATE standards (note: the standards may be provided by the customer)
  - Creating Batch Schedules – Triggered/Self-start
  - Incorporating Exception handling strategy
  - Imparting restart and recovery capabilities
- Creating semantic layer as per design document
- Creating Reports as per design document
- Publishing the reports
- Creating User Groups and Access Permissions according to the Design Document
- Adopting Standard Version Control Mechanisms
- Preparing Test Scripts for SIT


2.1: Data Warehouse Development Life Cycle
Construction (Contd...)
Creating Reports as per design document
Publishing the reports
Creating User Groups and Access Permissions according to the Design Document
Adopting Standard Version Control Mechanisms
Preparing Test Scripts for System Integration Testing (SIT)
IGATE

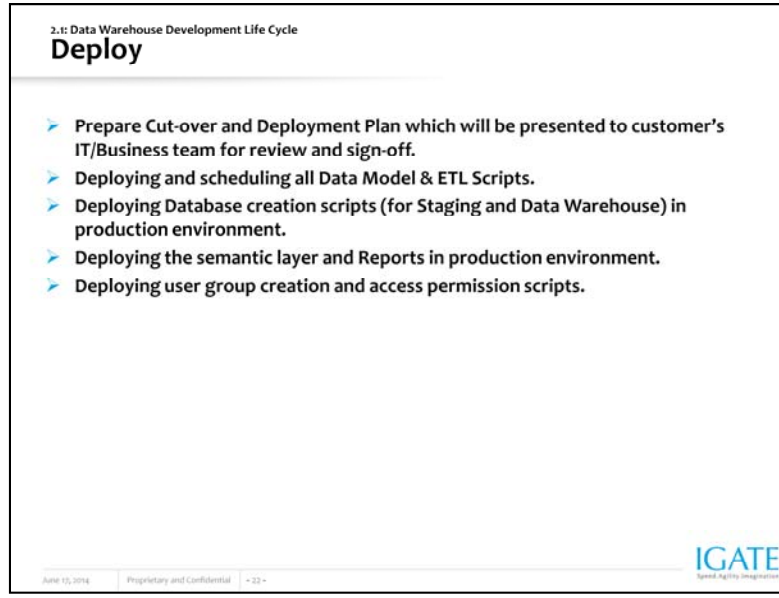


Task under warehouse designing:

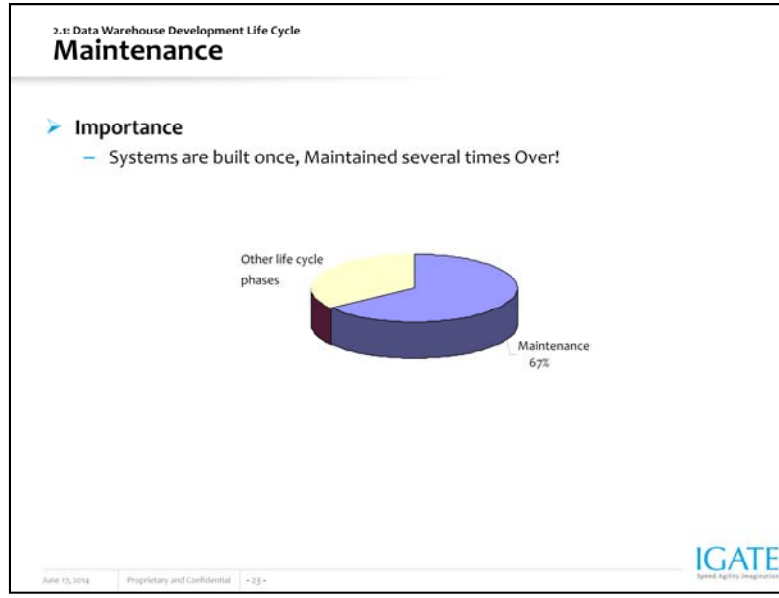
- Identifying the Sources
- Cleaning the Data
- Transforming the Data
- Designing Subsidiary Targets



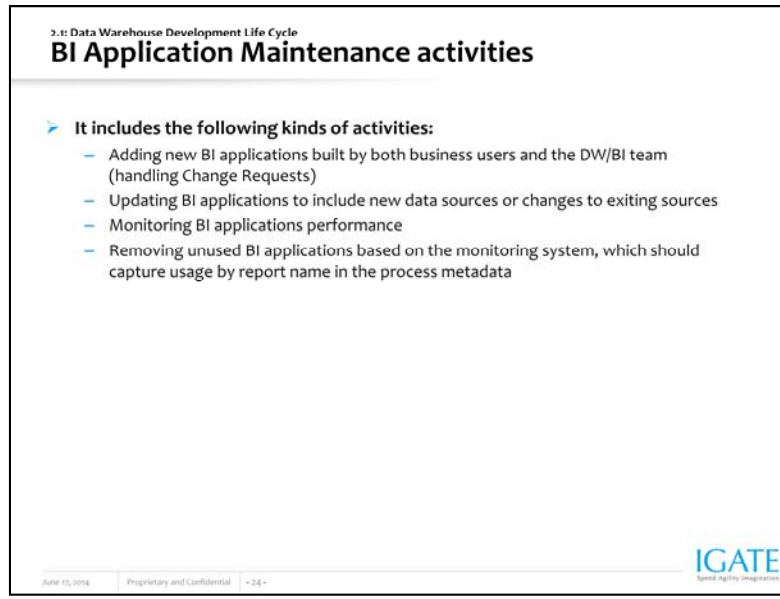
Following the principal that quality must be built into the system, rather than added on, organization follows an SDLC process in which appropriate user and regulatory requirements and design specifications are established and approved early in the process and defects are detected and corrected as part of the development activities. Comprehensive, documented unit and integration testing during development ensures the system is constructed correctly before it is installed for formal validation activities. Structural and functional testing documented during development does not need to be duplicated during formal validation testing, which will focus on system and user acceptance testing of the fully integrated system.



The scripts will be migrated to the production environment after validation and successful completion of user acceptance and subsequently will be deployed. Detailed activities:
Prepare Cut-over and Deployment Plan which will be presented to customer's IT/Business team for review and sign-off
Deploying and scheduling all Data Model & ETL Scripts
• Deploying Database creation scripts (for Staging and Data Warehouse) in production environment
• Deploying the semantic layer and Reports in production environment
• Deploying user group creation and access permission scripts



The importance of maintenance stems from the fact that it is only Change that is constant. A system building would happen only once, but it may need modifications many times over to keep it in use. It is estimated that 67% efforts are spent in maintenance. No wonder, the industry is now paying more attention to the ease of maintenance.



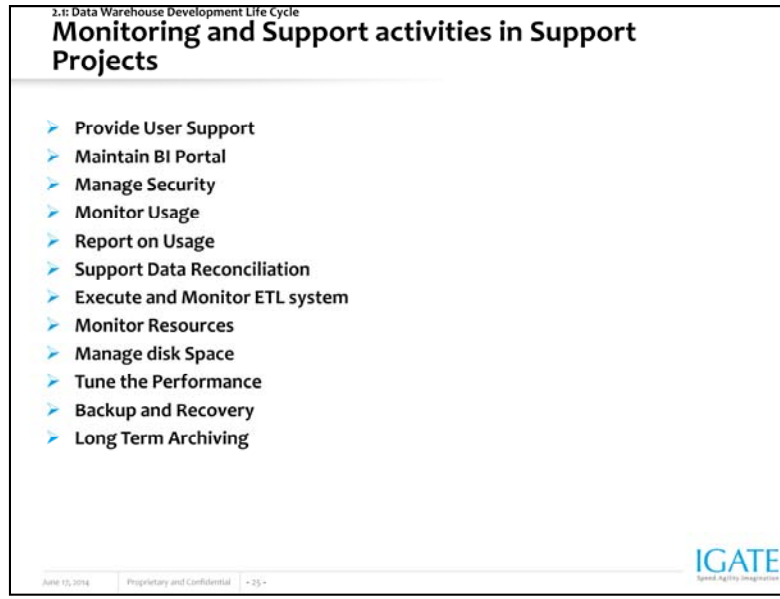
BI applications are not one time project. The initial set, and all subsequent addition, will need to be maintained and enhanced. This means someone will need to revisit the reports on a regular basis to verify their continued correctness and relevance in the organization. The team will also need resources to respond to requests for a additional reports and analyses.

Once the data warehousing system goes live, there are often needs for incremental enhancements. I am not talking about a new data warehousing phases, but simply small changes that follow the business itself. For example, the original geographical designations may be different, the company may originally have 4 sales regions, but now because sales are going so well, now they have 10 sales regions.

Deliverables

Change management documentation

Actual change to the data warehousing system



The task required to keep BI system operating in great shape are not difficult, but you need to plan and build for a maintainable system from the outset.

Primary tasks of Maintenance and support projects are as below:

Provide User Support:

BI system need to provide ongoing support to use its user community. In a three tiered support approach, the first tier is the website and self-service support; the second tier is your power users in the business groups; the third tier is front end people on the BI team

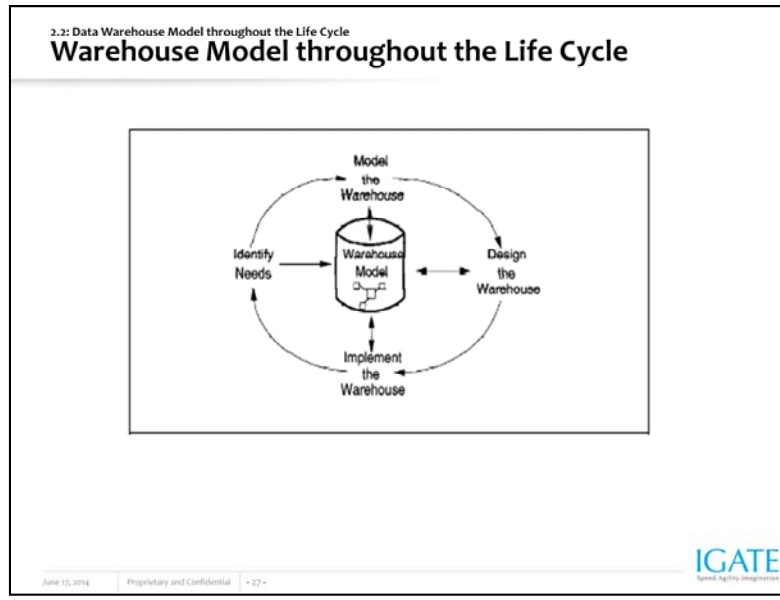
Maintain BI Portal:

- BI portal which is useful place to publish information about the BI system can have additional maintenance information such as:
- Data warehouse status
- Schedules of planned outages
- Clear warnings to users about problems in the system such as data quality issues.
- System's current operational status, including how many reports have been generated, etc
- **Manage Security:**

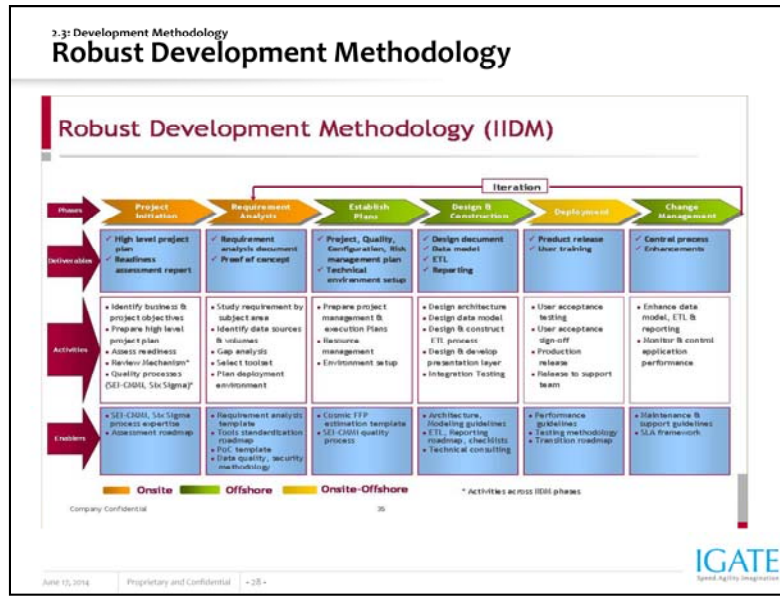
Most reporting tools have a user interface for managing roles and privileges. Security that's implemented in the database itself is usually handled by the BI security manager.

Backup and Recovery:
Ideally, back up the following databases after each load:
- Relational Data warehouse databases
- Staging databases
- Staging data in the file system
- Metadata databases

Long Term Archiving:
- Instead of betting on any single physical media technology, you should plan on periodic program of migrating and refreshing. Every three years, for instance, make sure that the data can be read and that it is stored on most current media of the day.
- Encapsulate your current application run time environment on a virtual machine. Archive that virtual machine image, and subject it to the migrate and refresh cycle.

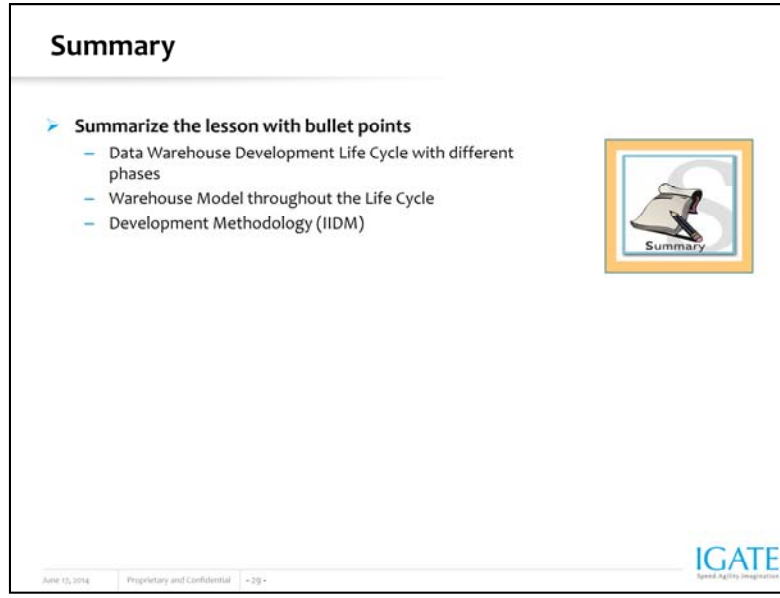


Once a data warehouse is implemented, usage of it will spawn new requests an requirements. This will start another cycle of development, continuing the iterative and evolutionary process of building the data warehouse. As you can see, the data model is a living part of a data warehouse. Through the entire life cycle of the data warehouse, the data model is both maintained and used The process of data warehouse modeling can be truly endless

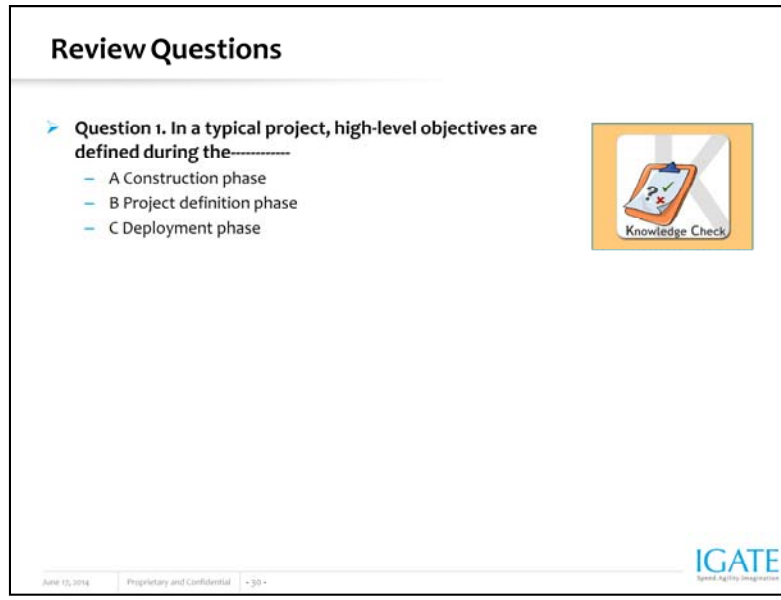


IIDM stands for Iterative Incremental Development Methodology.
Each development track has a specific deliverable which contributes to the BI project objectives:

- The ETL track will deliver loaded databases.
- The application track will deliver the reports, queries and ad hoc tools.
- The meta data repository will deliver the meta data.



Add the notes here.



Add the notes here.