

BI Software Development Lifecycle



Copyright © 2011 IGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of IGATE Corporation.

IGATE Corporation considers information included in this document to be Confidential and Proprietary.

BI Software Development Lifecycle

Document History

Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
15-June-09	0.1D		BI CDI team	Review
20-June-09	0.1		Rajita Dhumal	Content Creation
20-July-11	0.2D		Karthikeyan/Rajitha/Vandan a	iGATE Patni Integration
10-Sep-12	1.0		Karthikeyan	iGATE Format

June 12, 2014

Proprietary and Confidential

+ 2 +



Course Goals and Non Goals

➤ Course Goals

- BI Solutions Overview
- BI Project Execution Methodology

➤ Course Non Goals

- BI Testing for each layer
- BI Tools for each layer



Pre-requisites

- Data Warehousing Concepts

June 12, 2014

Proprietary and Confidential

+ 4 +



Intended Audience

- Developers
- Test Engineers
- Designers



June 12, 2014

Proprietary and Confidential

+ 5 +

IGATE
Speed. Agility. Imagination.

Day Wise Schedule

➤ Day 1

- Lesson 1: BI Project Execution Overview
- Lesson 2: BI Project Execution Methodology (Development)
- Lesson 3: Most Used Templates

June 12, 2014

Proprietary and Confidential

+ 6 +



Table of Contents

- **Lesson 1: BI Project Execution Overview**
 - 1.1. Software Project Types
 - 1.2. Execution Models (ON/OFF,ON,OFF)
 - 1.3. Typical Stakeholders
- **Lesson 2: BI Project Execution Methodology (Development)**
 - 2.1. Data Warehouse Development Life Cycle
 - 2.1.1. Manage the Project
 - 2.1.2. Define the Project
 - 2.1.3. Analysis
 - 2.1.4. Design
 - 2.1.5. Construction
 - 2.1.6. Deploy
 - 2.1.7. Maintenance
 - 2.2. Warehouse Model throughout the Life Cycle
 - 2.3. Development Methodology (IIDM)
- **Lesson 3: Most Used Templates**

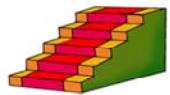
References

- **The Data Warehouse Lifecycle toolkit by Ralph Kimball**
- **Data Modeling Techniques for Data Warehousing by Chuck Ballard**
- **The Data Warehouse ETL toolkit by Ralph Kimball**



Next Step Courses (if applicable)

- BI Testing



June 12, 2014

Proprietary and Confidential

+ 9 +

Other Parallel Technology Areas

- Traditional SDLC

June 12, 2014

Proprietary and Confidential

+ 10 +



BI Software Development Lifecycle

Lesson 1: BI Project Execution Overview

June 17, 2018

Proprietary and Confidential

- 9 -



Lesson Objectives

- Introduction to Software Systems
- Software Project Types
- Execution Models
- Typical Stakeholders



IGATE
Speed. Agility. Imagination.

June 17, 2018

Proprietary and Confidential

- 2 -

I.O: Software Systems Overview

Introduction to Software Systems

- **Evolution of Software Systems**
 - Specialized to All Pervasive
- **Complexity Involved in Building Software Systems**
- **Need for a framework for building Software**
 - Software Engineering
- **Generic Phases of Software Engineering**
 - Definition or What/ Why Phase – Analysis Phase
 - Design – How Phase
 - Development - Implementation/Construction Phase
 - Testing Phase
 - Maintenance Phase

June 15, 2018 | Proprietary and Confidential | + 9 +

IGATE
Speed. Agility. Imagination.

Just as hardware and programming languages have evolved through several generations, so have software systems.

In the early days, the software systems were specialised problem solving tools. Today, they are present everywhere!

Programming was more or less an art then - there were few formal methods and the software world was virtually undisciplined. But software systems grew more and more complex and also critical. They also became time bound. This complexity came from various factors. For eg., the complexity of problem domain; or the difficulty in managing changes to requirements or the software building process.

To deal with this rising complexity & criticality, it became necessary to have a disciplined and systematic approach for software development. This framework for building software is provided by software engineering.

1.1: Software Project Types

Software Project Types

- **Development**
 - Translating Users' Needs into New Software Systems/ Product
- **Conversion / Migration**
 - Modifying Existing Software Systems to Different Environments
- **Maintenance**
 - Modifying Existing Software Systems to Correct/ Adapt/ Enhance
- **Support**
 - BI system need to provide ongoing support to use its user community

June 15, 2014 | Proprietary and Confidential | + 0 -

IGATE
Speed. Agility. Integration.

In the Development Projects, user's needs are translated into a New Software system or Product. For example, development of a new financial system.

In the Conversion Projects, modification of existing software is done to enable it to operate with similar functionality/ capability, but in a different environment. For example, conversion from C to C++, Port from Windows to Solaris etc.

In the Maintenance Projects, s/w system is modified to:

- a. Correct Faults (Corrective Maintenance)
- b. Modify software to accommodate changes to different environment (Adaptive Maintenance)
- c. Add functionality/ improve performance or other attributes (Enhancement)

Examples could include maintaining a financial application or a CAD application

There is also another kind of maintenance called as Preventive maintenance, where changes to system

are made so that they can be more easily corrected/ enhanced or adapted. One should note that regardless of the project type, similar life cycle activities - albeit in various measures – would exist.



June 15, 2014

Proprietary and Confidential

+ 5 +

IGATE
Speed. Agility. Integration.

Support projects are also considered as a different project type. Typical activities that comes under support projects are as follows: Provide User Support, Maintain BI Portal, Manage Security
Estimation is different for each of these project types.

1.2: Execution Models
Execution Models (On/Off, Off, On)

Onsite/Offshore	Offshore	Onsite
Is used for development / maintenance projects	Is used for maintenance / support projects	Client manages resources which could be single-slot or two-slot for time frame projects
The OPM works from the client site and coordinates project related aspects with the ODC and the client	Onsite presence is not required.	Connectivity to offshore is not available.
As nature of project involves intense collaboration or interaction on a continual basis, an onsite presence is maintained.		There are some legal limitations such as not to give access to offshore.

June 15, 2014 | Proprietary and Confidential | + 0 +

IGATE
Speed. Agility. Integration.

Execution methodology uses of commonly following three approaches:

- Onsite/Offshore
- Offshore
- Onsite

Based on the nature and scope of the projects, resources get allocated to it. Accordingly it could be single slot where-in only one resource manage the entire implementation or could be two slot or more.

1.2: Execution Models
Onsite/Offshore Model

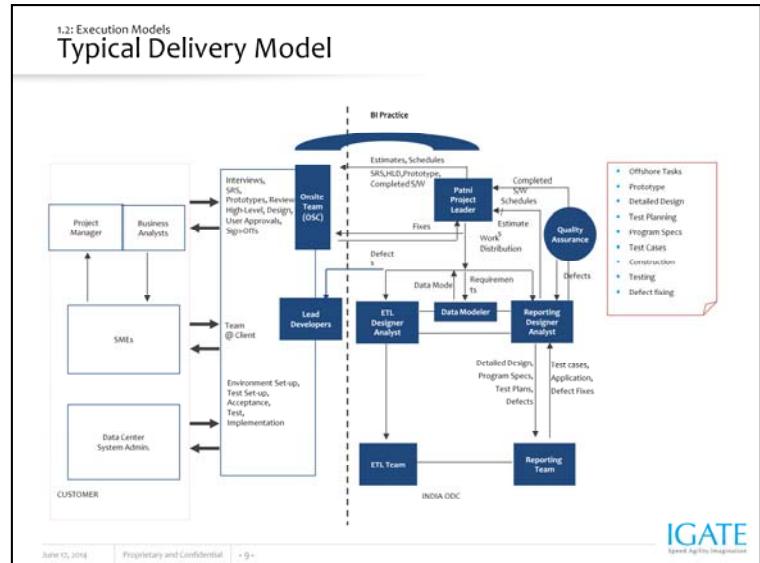
- Where the nature of the project involves intense collaboration or interaction on a continual basis or there is a customer specific requirement, an onsite presence is maintained
- The onsite manager works from the customer site and coordinates project related aspects with the ODC and the customer
- This onsite presence may involve a single resource or an entire project services team, depending on the nature, size, and complexity of the development initiative.

1.2: Execution Models

Onsite/Offshore Model (Contd...)

➤ **The important features of this approach are:**

- The approach requires an onsite coordinator (OSC) or an onsite project manager.
- The onsite component may include one or more resources depending upon the size and complexity of the project.
- A considerable amount of the work effort can be transitioned to the offshore facility. That transition can result in a significant cost benefit to the customer and highest usage efficiency of resources.
- OSC can interact directly with the clients to accommodate any changes, if there are any, so as to minimize/eliminate last minute changes.



The project execution team at INDIA ODC handles following activities: Offshore Tasks, Prototype, Detailed Design, Test Planning, Program Specs, Test Cases, Construction, Testing, Defect fixing, etc

Onsite work is handled by the highly qualified consultants present locally.

These consultants are engaged in continuous interaction with the clients directly in understanding their requirements/specifications, communicating on functionalities and best business practices available, as well as formulating solutions to unique business needs of the client.

Activities carried out by Onsite team includes Interviews, SRS(System Requirement Specification)/RAD(Requirement Analysis Document), Prototypes, Review, High-Level Design, User Approvals, Sign-offs from customer etc.

1.3. Typical Stakeholders

Stakeholders

- A stakeholder is anyone who is a direct user, indirect user, manager of users, senior manager, operations staff member, the "gold owner" who funds the project

Business Users ➤ Dashboard Users ➤ Reporting Users - Static Report Users - Ad-hoc Report Users	IT Team ➤ Data Owners / Data Analyst ➤ OLTP IT Team ➤ OLAP IT Team ➤ Admin Team (DBA, infrastructures, etc) ➤ Support Team (for above mentioned stakeholders)	Help Desk Is maintained for the Support projects
--	--	---

IGATE
Speed. Agility. Integration.

June 17, 2014 | Proprietary and Confidential | + 10 +

Stakeholders

- a project stakeholder is anyone who is a direct user, indirect user, manager of users, senior manager, operations staff member, the "gold owner" who funds the project
- supports (help desk) staff member, your program/portfolio manager, developers working on other systems that integrate or interact with the one under development,
- maintenance professionals potentially affected by the development and/or deployment of a software project.

Project success often requires a greater level of involvement by project stakeholders – senior management needs to publicly and privately support your project, operations and support staff must actively work with your project team towards making your production environment ready to accept your system, other system teams must work with yours to support integration efforts, and maintenance developers must work to become adept at the technologies and techniques used by your system.

Business people, such as direct users and their managers, aren't the only stakeholders of a project. As you know there is a wide range of people potentially affected by a new system, therefore to succeed you must understand and then synthesize their requirements into a cohesive vision. This is one of the things that makes software development hard – each project stakeholder will have their own requirements, their own vision, and their own priorities – but it also makes software development fun.

Summary

- Software Project Types
- Execution Models
- Typical Stakeholders



IGATE
Speed. Agility. Integration.

June 15, 2014

Proprietary and Confidential

- 11 -

Add the notes here.

Review Questions

- Question 1 A project stakeholder is anyone who is a direct user, indirect user, who funds the project
 - True/ False?
- Question 2 _____ project type is used for translating Users' Needs into New Software Systems/ Product



Add the notes here.

BI Software Development Lifecycle

Lesson 2: BI Project Execution Methodology (Development)

June 12, 2014

Proprietary and Confidential

- 1 -



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)



2.1: Data Warehouse Development Life Cycle

Data Warehouse Development Life Cycle

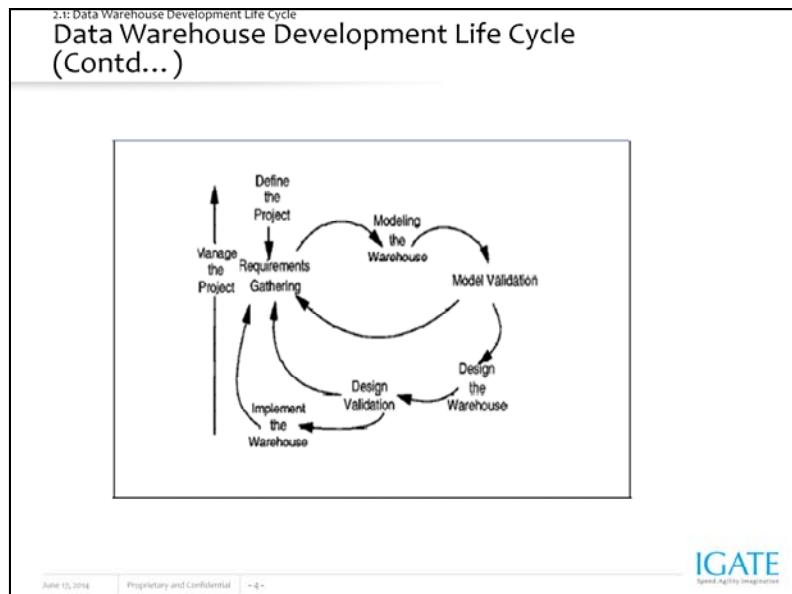
- Data warehouse projects must focus on the needs of the business.
- While data warehousing is an ongoing process, each implementation project should have a finite cycle with a specific beginning and end.
- Whereas all projects have some degree of iteration, data warehouse projects take iteration to the extreme to enable fast delivery of portions of a warehouse.
- Thus portions of a data warehouse can be delivered while others are still being developed.

June 15, 2014 | Proprietary and Confidential | - 3 -

IGATE
Speed. Agility. Integration.

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

Manage the Project

- Why is project management for a data warehouse different than most other applications?
- Data warehouses are ever changing, dynamic. This is what makes project management for a data warehouse so unique and challenging

June 15, 2014 | Proprietary and Confidential | + 5 +

IGATE
Speed. Agility. Integration.

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.

2.1: Data Warehouse Development Life Cycle

Define the Project

- 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.
- 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.

June 15, 2014 | Proprietary and Confidential | + 6 +

IGATE
Speed. Agility. Integration.

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.

2.11 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

Why???



What???

IGATE
Speed. Agility. Integration.

June 15, 2014 | Proprietary and Confidential | + 8 +

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

2.1 Data Warehouse Development Life Cycle

Requirements Gathering

- There are many methods for deriving business requirements
- In general, these methods can be placed in one of two categories:
 - source-driven
 - user-driven

IGATE
Speed. Agility. Integration.

June 15, 2014 | Proprietary and Confidential | + 10 +

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?

2.1 Data Warehouse Development Life Cycle Source-Driven Requirements Gathering

- Is a method based on defining the requirements by using source data in production operational systems.
- This is done by analyzing an ER model of source data if one is available or the actual physical record layouts and selecting data elements deemed to be of interest.
- The result of the source-driven approach is to provide the user with what you have.

June 15, 2014

Proprietary and Confidential

- 11 -

IGATE
Speed. Agility. Integration.

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.

2.1: Data Warehouse Development Life Cycle
User-Driven Requirements Gathering

- Is a method based on defining the requirements by investigating the functions the users perform.
- This is usually done through a series of meetings and/or interviews with users.
- For a full-scale data warehouse, it would be worth to use the source-driven approach to break the project into manageable pieces, which may be defined as subject areas.
- The user-driven approach could then be used to gather the requirements for each subject area.

June 15, 2014 | Proprietary and Confidential | - 12 -

IGATE
Speed. Agility. Integration.

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.

2.1: Data Warehouse Development Life Cycle

Design

➤ **The activities will include:**

- Finalizing the Logical Data Model
- Finalizing the Physical Data Model (Tables, Views, Keys, indexes, partitions etc.) for the Data Warehouse and Staging Area.
- Finalizing ETL strategy & design.
- Finalizing detailed design of semantic layer (if any) for reports.
- Finalizing detailed design of reports.
- Finalizing User Groups and Access Permissions for reports.

HOW???



June 17, 2014 | Proprietary and Confidential | 13 | **IGATE**
Speed. Agility. Ingenuity.

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.)

2.1 Data Warehouse Development Life Cycle
Design (Contd...)

➤ **Importance**

- Gives a blueprint for system development
- Overall Quality: Good design helps
 - To build stable software
 - To reduce coding & testing time
 - To easily maintain system

June 15, 2014 | Proprietary and Confidential | - 14 -

IGATE
Speed. Agility. Integration.

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.

2.1: Data Warehouse Development Life Cycle Modeling the Data Warehouse

- Modeling the target warehouse data is the process of translating requirements into a picture along with the supporting metadata that represents those requirements. It is designing the flow of data graphically.
- As soon as some initial requirements are documented, an initial model starts to take shape.
- At the end of the modeling phase, you have a complete picture of the requirements.

June 15, 2014 | Proprietary and Confidential | - 16 -



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.

2.1: Data Warehouse Development Life Cycle
Validating the Model

- **The purpose of validating your model with the user:**
 - it serves to confirm that the model can actually meet the user requirements
 - a review should confirm that the user can understand the model

June 15, 2014 | Proprietary and Confidential | - 17 -



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.

2.1: Data Warehouse Development Life Cycle

Construction

➤ **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
- Creating semantic layer as per design document

June 15, 2014 | Proprietary and Confidential | +15 +

IGATE
Speed. Agility. Integration.

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)

2.1: Data Warehouse Development Life Cycle

Design the Warehouse

- Once a model is created and validated, it is analyzed to determine the best way to physically implement it.
- One area where design can impact performance is renormalizing, or snowflaking, dimensions.
- This decision should be made based on how the specific query tools you choose will access the dimensions.

June 15, 2014 | Proprietary and Confidential | +20 +

IGATE
Speed. Agility. Integration.

Task under warehouse designing:

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

2.1: Data Warehouse Development Life Cycle
Validating the Design

- Comprehensive, documented unit and integration testing during development ensures the system is constructed correctly before it is installed for formal validation activities

June 15, 2014 | Proprietary and Confidential | - 21 -



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.

2.1: Data Warehouse Development Life Cycle

Deploy

- 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.

June 15, 2014 | Proprietary and Confidential | + 22 +

IGATE
Speed. Agility. Integration.

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

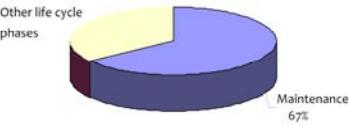
- Deploying user group creation and access permission scripts

2.11 Data Warehouse Development Life Cycle

Maintenance

➤ **Importance**

- Systems are built once, Maintained several times Over!



Phase	Percentage
Maintenance	67%
Other life cycle phases	25%
Change	8%

June 15, 2014 | Proprietary and Confidential | +23+

IGATE
Speed. Agility. Integration.

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.

2.11 Data Warehouse Development Life Cycle

BI Application Maintenance activities

➤ It includes the following kinds of activities:

- Adding new BI applications built by both business users and the DW/BI team (handling Change Requests)
- Updating BI applications to include new data sources or changes to existing sources
- Monitoring BI applications performance
- Removing unused BI applications based on the monitoring system, which should capture usage by report name in the process metadata

June 15, 2014 | Proprietary and Confidential | +24+

IGATE
Speed. Agility. Integration.

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

2.11 Data Warehouse Development Life Cycle

Monitoring and Support activities in Support Projects

- Provide User Support
- Maintain BI Portal
- Manage Security
- Monitor Usage
- Report on Usage
- Support Data Reconciliation
- Execute and Monitor ETL system
- Monitor Resources
- Manage disk Space
- Tune the Performance
- Backup and Recovery
- Long Term Archiving

IGATE
Speed. Agility. Integration.

June 15, 2014 | Proprietary and Confidential | +25+

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.

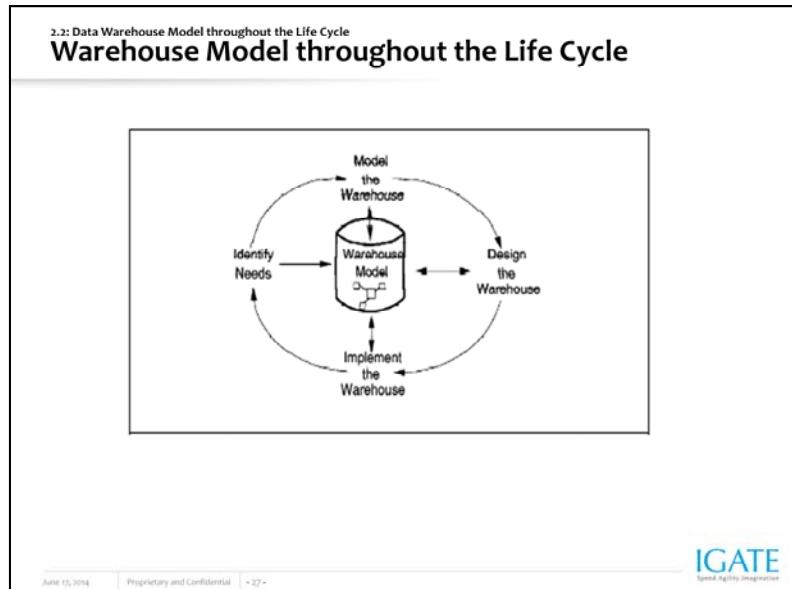
Backup and Recovery:

Ideally, back up the following databases after each load:

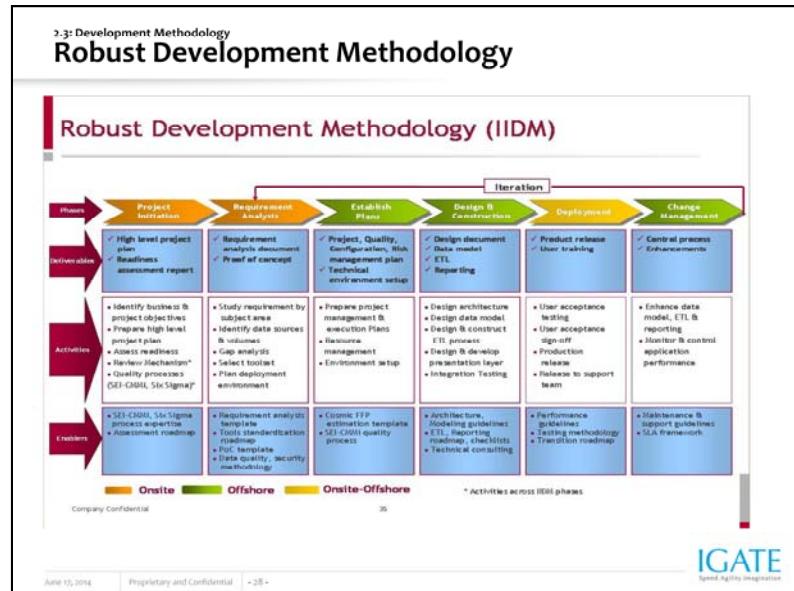
- Relational Data warehouse databases
- Staging databases
- Staging data in the file system
- Metadata databases

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 and 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.

Summary

➤ Summarize the lesson with bullet points

- Data Warehouse Development Life Cycle with different phases
- Warehouse Model throughout the Life Cycle
- Development Methodology (IIDM)



IGATE
Speed. Agility. Integration.

June 15, 2014 | Proprietary and Confidential | +29+

Add the notes here.

Review Questions

- Question 1. In a typical project, high-level objectives are defined during the-----
- A Construction phase
 - B Project definition phase
 - C Deployment phase



June 15, 2014 | Proprietary and Confidential | +30 +

IGATE
Speed. Agility. Integration.

Add the notes here.

BI Software Development Lifecycle

Lesson 3: Most Used Templates

June 12, 2014

Proprietary and Confidential

- 3 -



Lesson Objectives

- Most Used Templates in BI Systems



June 12, 2014

Proprietary and Confidential

- 2 -

IGATE
Speed. Agile. Imagination

Most used templates

- Requirements Specs
- HLD and LLD (ETL and OLAP)
- Mapping Specs
- Report Specs
- Test Plan
- Test Cases and Results
- QTS
- Issue Log

Report Request Form

Report Request Form

Requested by _____ Report Specification Name: _____ REPORT NAME: _____	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Report Check Out: <input type="checkbox"/> Level 1 <input checked="" type="checkbox"/> Non-Request <input type="checkbox"/> Masked/Speculative <input type="checkbox"/> Level 2 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Level 3 <input type="checkbox"/> <input type="checkbox"/>	Task ID: _____ Assigned To: _____ Status: _____ Last Report Name: _____ Date Assigned: _____ Last Reviewed: _____																				
Basic Functionality & Logic of the Report																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Prompt Name</th> <th>Prompt Type</th> <th>Description of prompt (e.g., month, percentage, date, item mapping for prompt columns)</th> <th>Report filter status</th> <th>If use last selection for the previous report</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>			Prompt Name	Prompt Type	Description of prompt (e.g., month, percentage, date, item mapping for prompt columns)	Report filter status	If use last selection for the previous report															
Prompt Name	Prompt Type	Description of prompt (e.g., month, percentage, date, item mapping for prompt columns)	Report filter status	If use last selection for the previous report																		
Report Header:																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">Per:</td> <td style="width: 90%;">Last Generated/Last Modification - Client & Credential Configuration - Client & User - Categories assigned to it (if any other category) - Categories assigned to it (if any other category) - Sub-Prompts - Used to build the header & footer of application - Contains target report & contains data items</td> </tr> </table>			Per:	Last Generated/Last Modification - Client & Credential Configuration - Client & User - Categories assigned to it (if any other category) - Categories assigned to it (if any other category) - Sub-Prompts - Used to build the header & footer of application - Contains target report & contains data items																		
Per:	Last Generated/Last Modification - Client & Credential Configuration - Client & User - Categories assigned to it (if any other category) - Categories assigned to it (if any other category) - Sub-Prompts - Used to build the header & footer of application - Contains target report & contains data items																					
Report Details:																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Fields needed (COLUMNS in Report)</td> <td style="width: 33%;">Field Name: _____ Data Item: _____ Data Item: _____ Data Item: _____</td> <td style="width: 33%;">Field Name: _____ Data Item: _____ Data Item: _____ Data Item: _____</td> </tr> </table>			Fields needed (COLUMNS in Report)	Field Name: _____ Data Item: _____ Data Item: _____ Data Item: _____	Field Name: _____ Data Item: _____ Data Item: _____ Data Item: _____																	
Fields needed (COLUMNS in Report)	Field Name: _____ Data Item: _____ Data Item: _____ Data Item: _____	Field Name: _____ Data Item: _____ Data Item: _____ Data Item: _____																				
Report Footer: (If applicable)																						
Additional Requirements: - Max conditions of branching, repeated logic, precision requirements etc. - Report Layout (attach sample layout):																						

Information about Report Layout as well the report field's needed is documented as a part of the Report specification template.

This template is used to maintain status of the issued generated.

Assigned to: The person to whom the issues has been given to complete or follow up on

Actions: What is to be done to complete the issue

This QTS template is used to maintain the status of the queries raised.

- Enter all requirements related queries raised throughout project duration in this sheet to keep all queries at one place.
 - Insert queries raised and answered in proposal phase as well into this sheet, when the project begins.
This will ensure all replies and assumptions made in proposal phase are known to the development team.
 - Projects send/receive updated query tracking sheet to/from client/OSC. Only the latest version of the sheet should be maintained at designated place (to avoid confusion/rework).
The designated location should be pointed to by Projects' PML(Project Master Locator).
 - Projects should request client/OSC to enter their queries and replies also in this sheet. Even if client does not enter,
Projects should keep this sheet updated for important queries (preferably all queries), if queries are received by mails or documents.
 - The status of all pending queries should be taken in the project meetings and PSR (Project Status Report).
PL should escalate to OSC/Client for open queries

Test Cases								
Test Cases for <Functionality Title>								
Test Case #	Objective	Precondition	Steps	Expected results	Actual Results	Passed (Y / N)	Defect #	Remark
1.0	To verify the <brief description of purpose of the test case>							
1.1	<Objective of case 1>		<Describe the steps involved in testing the test case>					
1.2	<Objective of case 2>							
2.0	To verify the <brief description of purpose of the test case>							
2.1	<Objective of case 1>							
2.2	<Objective of case 2>							

 IGATE
Speed. Agility. Imagination.

The test case template is used to maintain the details of the test cases carried out. For every step, expected results and Actual results are maintained with the Status output as PASSED (YES) or PASSED(NO).

Precondition: Any prerequisite to be satisfied before execution of the test case

ETL Low Level Design Template

Low Level Design Template

This document compares Informatica PA versions for Dashboard & Chart functionality

DOCUMENT HISTORY

Date	Revision	Change
	0.01D	Created Initial Draft
26-Dec-05	1.0	Baselined

DOCUMENT CONTROL INFORMATION

Prepared By Kranti Modi

Maintained By BI CoE

All suggestions, comments, change requests and request for copies should be addressed to the BI-CoE Group.

Approved By BI CoE

Reference

IGATE
Speed Agility Imagination

June 15, 2014

Proprietary and Confidential

- 8 -

Summary

- Most Used Templates in BI Systems



IGATE
Speed. Agility. Imagination.

Add the notes here.