Previews of TDWI course books are provided as an opportunity to see the quality of our material and help you to select the courses that best fit your needs. The previews can not be printed.

TDWI strives to provide course books that are content-rich and that serve as useful reference documents after a class has ended.

This preview shows selected pages that are representative of the entire course book. The pages shown are not consecutive. The page numbers as they appear in the actual course material are shown at the bottom of each page. All table-of-contents pages are included to illustrate all of the topics covered by a course.

TDWI Data Integration Testing

Ensuring Quality for ETL and Data Consolidation

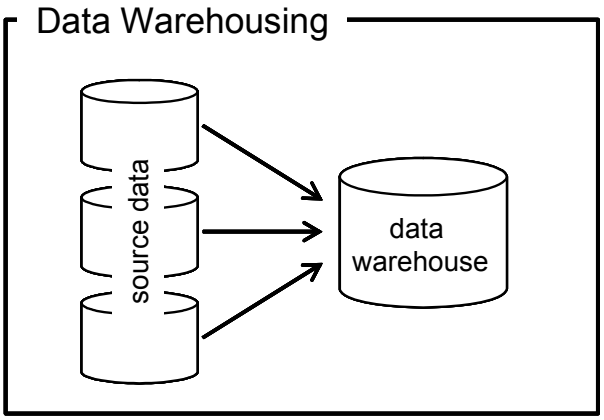
TABLE OF CONTENTS

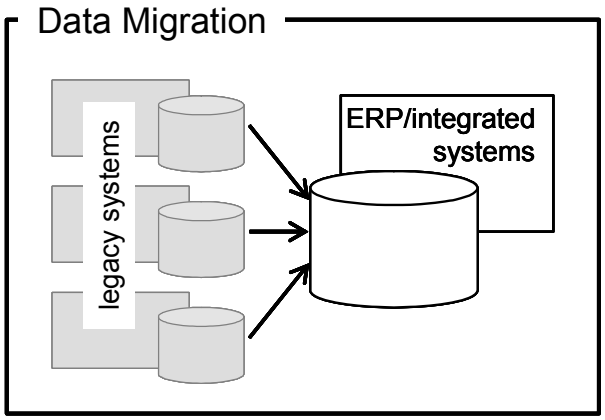
Module 1

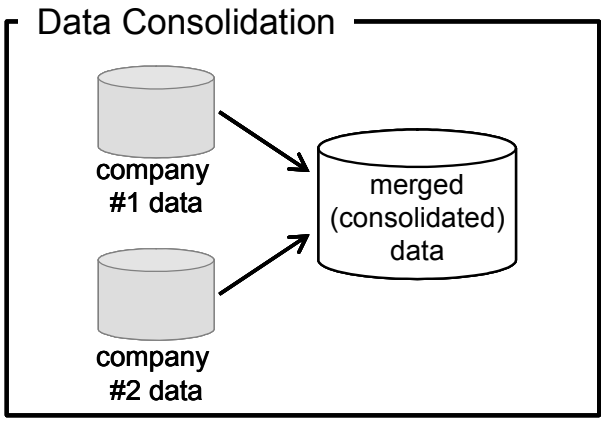
Concepts and Terminology

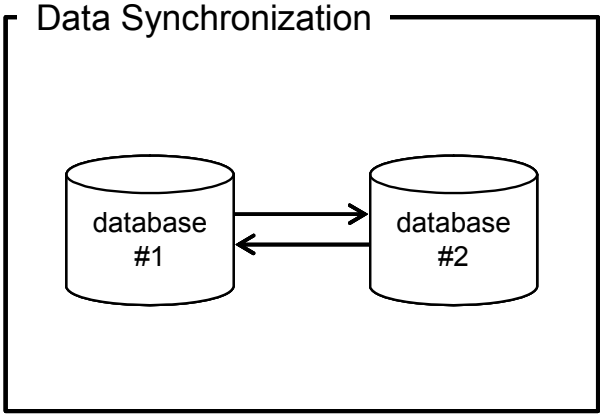
Data Integration Terms and Concepts

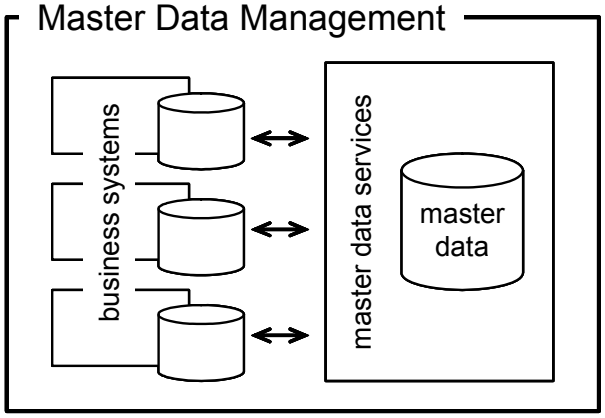
Processes and Projects











Data Integration Terms and Concepts

Processes and Projects

DATA WAREHOUSING

Data warehousing is the set of processes for collecting, storing, and managing data in a data warehouse. A data warehouse is a collection of integrated data designed to meet needs for end-user access, reporting, and business analysis. Bill Inmon's original definition of a data warehouse describes it as "a subject-oriented, integrated, time-variant, non-volatile collection of data" to meet these needs. Data warehousing involves extensive data transformation to achieve the desired properties of subject-orientation, integration, time-variance, and stability.

DATA MIGRATION

Data migration is the transfer of data between databases and computer systems. It is needed when changing or upgrading computer systems. Migration most frequently occurs when replacing legacy systems with ERP or other updated solutions. Data transformation for conversion of formats is almost always a part of data migration, even when upgrading or replacing a single system. When multiple target systems are highly integrated more extensive data transformation is needed.

DATA CONSOLIDATION

Data consolidation combines and integrates data from disparate sources. Consolidation is becoming increasingly common in today's business environment of mergers and acquisitions. When companies merge they combine their assets including their data. Data consolidation typically includes extensive transformation to achieve common definition, format, and structure for all of the data.

DATA SYNCHRONIZATION

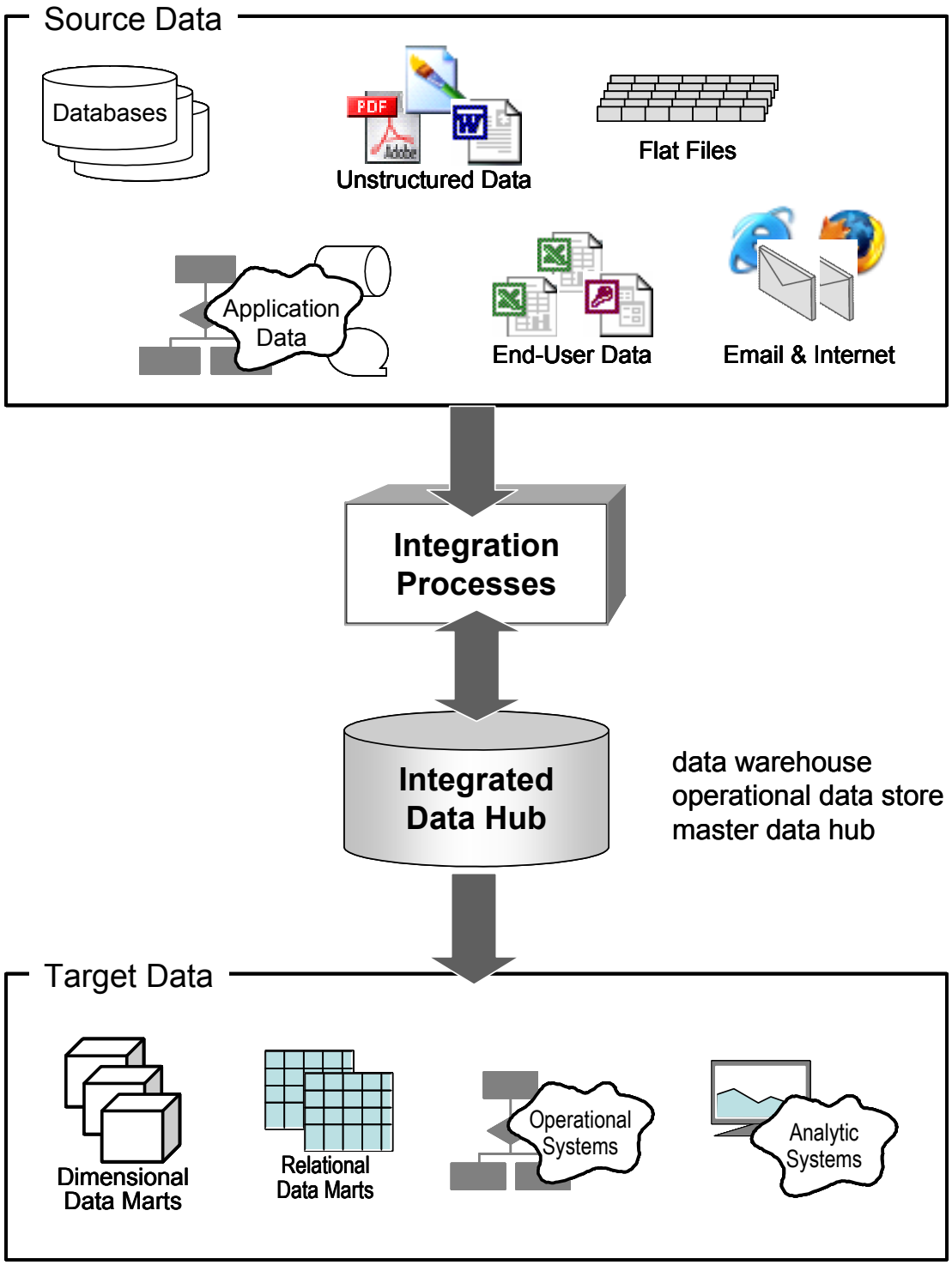
Data synchronization is performed to achieve consistency of overlapping data that is stored separately in multiple databases. Synchronization may be needed when multiple systems manage related data independently, when different organizations collect and store overlapping data, or when global data is widely distributed and transactions are applied locally. When global data standards are imposed (i.e., distributed databases) then transformation applies the standards to all of the data. Where global data standards are not applied, then data transformations may be unique to each database allowing for logical integration with local and physical independence.

MASTER DATA MANAGEMENT

Master data management (MDM) is similar to data synchronization in achieving consistency of overlapping data stored in multiple systems. MDM is specifically oriented to synchronization of enterprise reference data – customer, product, etc. – to provide consistency of reporting an analytics and to enable regulatory compliance. MDM technology goes beyond simple data movement to include master data services.

Data Integration Terms and Concepts

Architectures and Technologies – Integration Hub



Data Integration Terms and Concepts

Architectures and Technologies - Integration Hub

REPOSITORY OF INTEGRATED DATA

Data integration with hub-and-spoke architecture transforms data to be placed into an integrated data hub that becomes the point-of-access for consumers of integrated data. The data hub physically stores the integrated data in a single location and involves both processing and the database. Common hub-and-spoke implementation examples include:

- A data warehouse that feeds integrated data for dependent data marts.
- An operational data store (ODS) that supplies integrated data to the operational and reporting systems of the business.
- An enterprise-wide collection of integrated data that is accessed and used by analytic applications – a data warehouse, ODS, or combination of the two.

Integration hubs are typically implemented with batch processing to transform the data resulting in high-latency data, or with “fast batch” resulting in low-latency or near real-time data.

Testing Terms and Concepts

Processes and Testing

Table 1 - Manufacturing vs. Data Quality

Process Components	Manufacturing	Data Quality
Input	Raw Materials	Raw Data
Process	Materials Processing	Data Processing
Output	Physical Products	Data Products

Table 2: Data Integration and Data Quality Processes

Inputs: Raw Data	Actions	Outputs: Data Products
List of customers	Validation	Valid customer list
Complete list of invoices due	Sorting	Invoices sorted by date
Sales from all stores	Integrating	Total store sales
Prospect address list	Standardizing/Cleansing	Cleansed address list

Testing Terms and Concepts

Processes and Testing

DATA INTEGRATION AND DATA QUALITY ARE PROCESSES

Both data integration and data quality are processes – sequences of activities that receive inputs and produce value-added outputs. Process concepts are important testing considerations because processes offer several testing opportunities. You may, for example, choose to test the quality of inputs, the quality of activities performed, the quality of intermediate results produced by activities, and the quality of process outputs or finished goods.

DATA QUALITY PROCESSES

Quality is generally thought of as absence of defects. It is, however, somewhat more complex than simply being defect-free. Both absence of undesirable characteristics and presence of desirable characteristics are necessary to achieve quality in a product.

Undesirable and desirable characteristics, though, are rather vague concepts. To make them more concrete consider yet another definition: quality is *suitability to purpose*. A product is of high quality when it is well matched to its stated purpose and of low quality when it fails to fulfill that purpose.

In business intelligence and data warehousing, the purpose of each product is determined by its role in satisfying business requirements and meeting user expectations.

QUALITY ASSURANCE PROCESS

Quality assurance is a system of activities whose purpose is to provide to the producers and users of a product the assurance that it meets defined standards of quality with a high level of confidence. Quality assurance examines a product before its release or deployment to expose undesirable qualities – defects, deficiencies – and to ensure that desirable qualities are present.

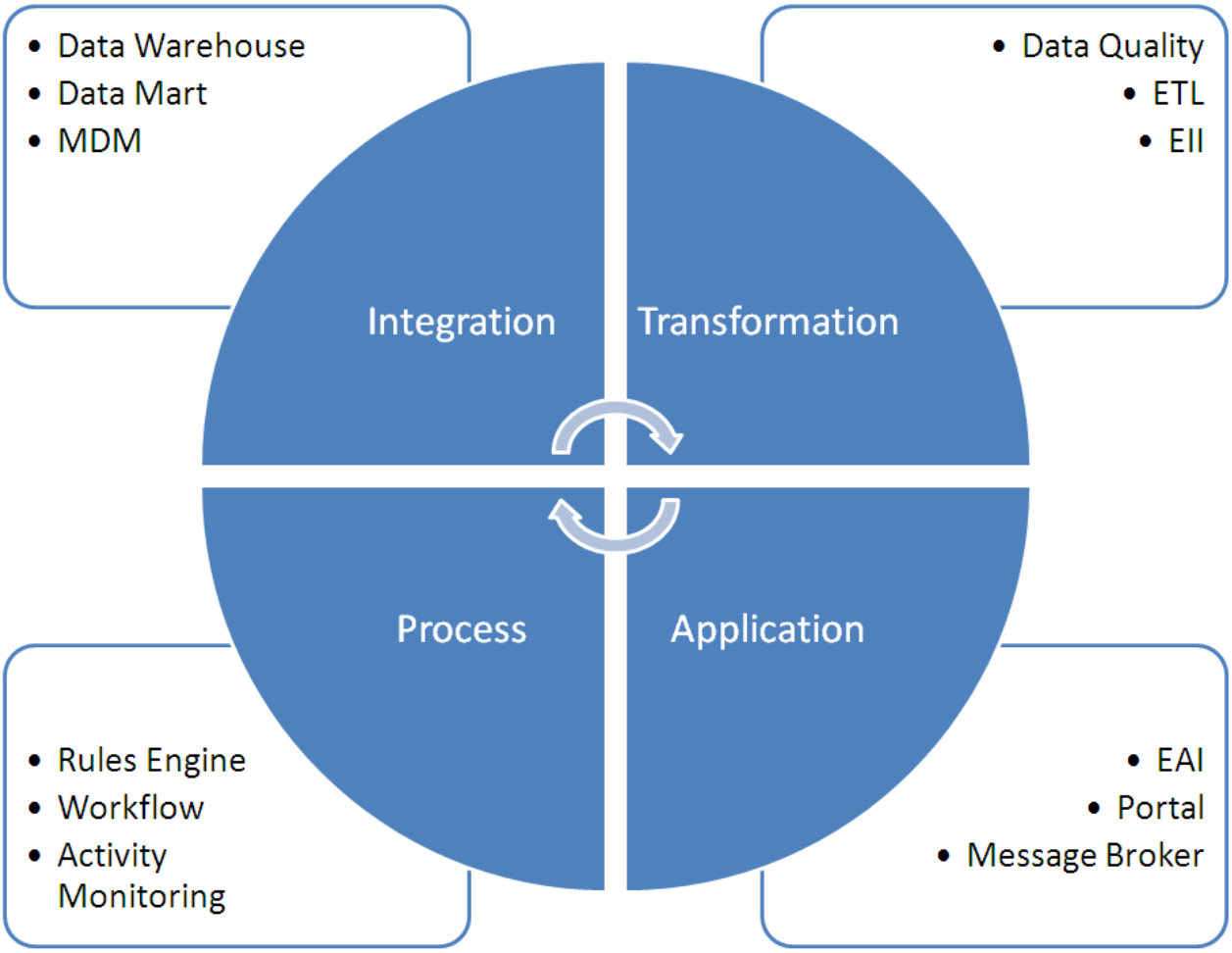
Module 2

Data Integration and Quality Models

This page intentionally left blank.

A Data Integration Framework

Four Dimensions of Data Integration



A Data Integration Framework

Four Dimensions of Data Integration

A FRAMEWORK While there are many dimensions to data integration, there are at least four core components including Integration, Transformation, Application, and Process

INTEGRATION One dimension of data integration is the requirement to consolidate, combined or otherwise integrate the data from disparate systems into a cohesive data store. Examples of target data structures include the data warehouse, data mart, or Master Data Management (MDM))

TRANSFORMATION An aspect of data integration is the necessity to cleanse the data being sourced and the propagate that data into a target data store. Therefore, data quality is a critical component of the transformation dimension. Other components include the traditional Extraction, Transformation and Loading (ETL) or ELT. A final example of transformation and propagation of data is the use of an Enterprise Information Integration (EII) technology.

APPLICATION While integration of data into a target data stores is often pursued, there are also requirements to source and propagate data between disparate applications. Enterprise Application Integration (EAI) technology is often cited as common platform to integrate data between applications. A message broker is another variation of technology that supports the event-based movement of data between applications. Another example of application integration revolves around Portal or Internet based solutions.

PROCESS Although there are several examples of data integration addressing process requirements, three common forms include the use of Business Rules Engines, Workflow Management, or Activity Monitoring.

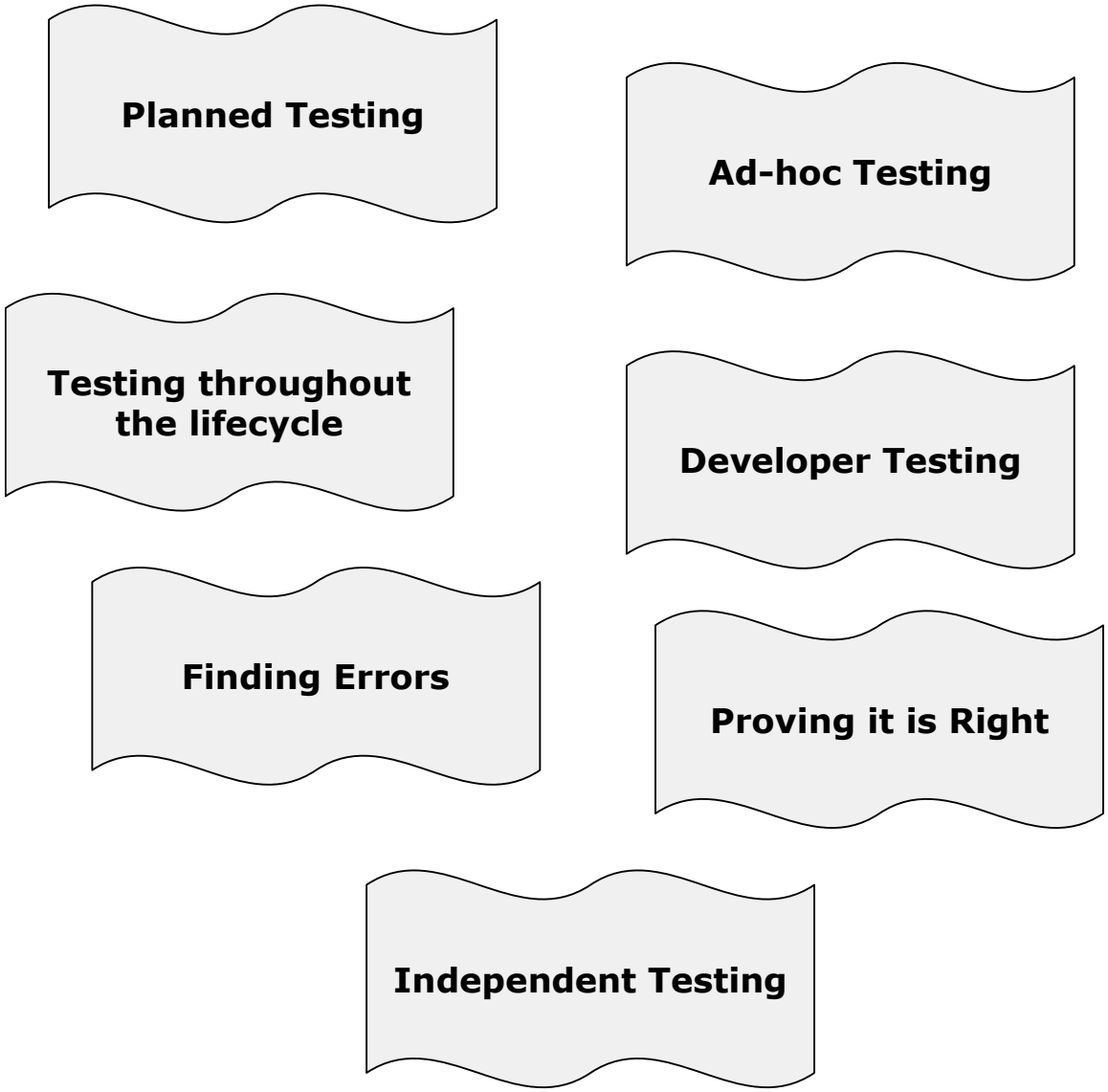
Module 3

Testing Practices and Techniques

This page intentionally left blank.

Testing Practices

Best Practices and Common Mistakes



Testing Practices

Best Practices and Common Mistakes

PLANNED TESTING VERSUS AD HOC TESTING

The largest challenge with the testing cycle is trying to complete the planned testing within the timeline, the projected costs, and with the allocated resources. Requirements and the associated test cases should be ranked and categorized to ensure critical tests are completed. Those requirements that are deemed less critical may be addressed through ad-hoc or de-facto testing. An example of a critical test would include the unit test of all new or changed code.

Another aspect to ad-hoc testing is the unplanned nature of the test. Ad-hoc testing can identify issues that may only arise through a certain use of the data.

FINDING ERRORS AND TESTING TO "PROVE THAT IT IS RIGHT"

Testing with a certain outcome in mind can be a common mistake in the testing approach. It is practical to test a product to ensure that it functions as expected; however, to fully test a product, the testing approach should include testing the unexpected, the unplanned, and scenarios that possibly could occur. A solid testing approach will include "proving that it is right" but also ensuring the product can function under unexpected circumstances. Unexpected circumstances could include an increase in data volume, receiving different data types, or an increase in user activity.

TESTING THROUGHOUT THE LIFE CYCLE

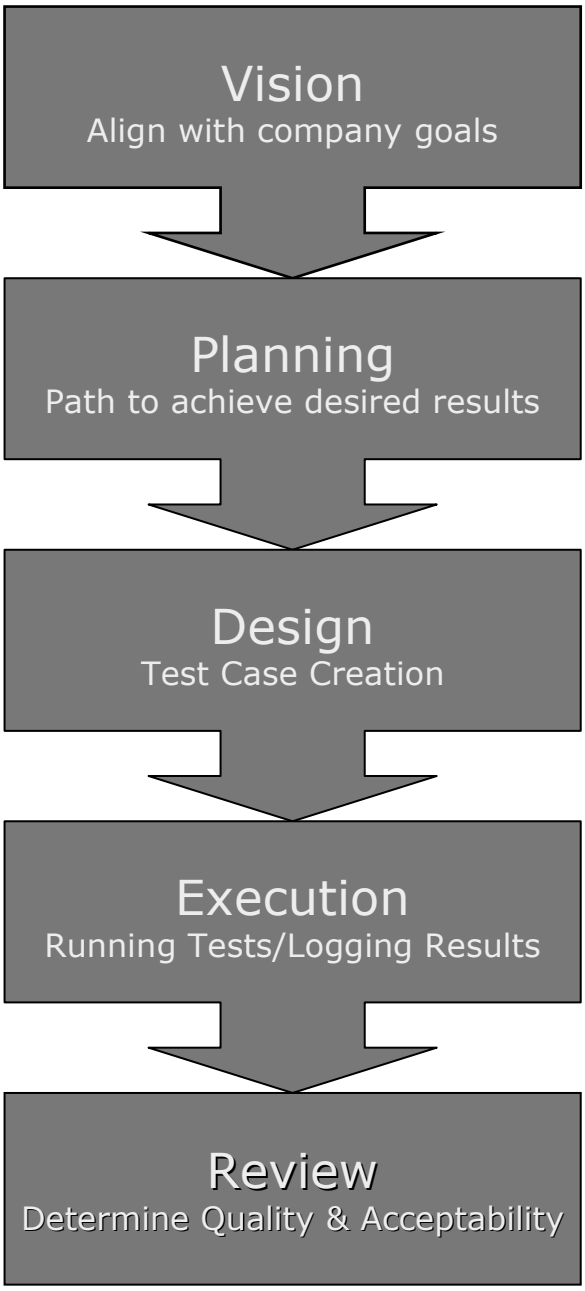
The testing process only finds defects it does nothing to correct them. To ensure the defect rate is as low as possible and issues are not caught late in the life cycle, testing at each phase to ensure completeness, correctness, and the overall quality is present is considered a best practice. Defects caught late in the lifecycle will impact the project more than if caught earlier in the lifecycle. Imagine having to re-design the ETL process due to a defect discovered in unit testing.

DEVELOPER TESTING VS. INDEPENDENT TESTING

Unit testing by the developer who created the product is a widely accepted practice. Testing beyond this scope by a developer could lower the quality of the test due to the lack of objectivity. Having independent testing as part of the process ensures a thorough and unbiased testing approach is completed.

Testing Practices

Structured Testing



Testing Practices

Structured Testing

VISION

All efforts have an end result in mind. The purpose behind creation of a data integration system is typically due to a larger effort connected to a vision. The corporate "vision" outlines where a corporation wants to be over a specific time period. The vision is used to govern new efforts a corporation undertakes to ensure that a corporation stays on course. Structured testing would include validating that the data integration system effort aligns with the overall vision. This is the first step in structured testing.

PLANNING

Planning is the next step in supporting the vision and achieving the target goal of the data integration system. The framework of the plan would include outlining the path to achieve the desired results. A structured test plan outlining the testing activities for the phases of requirements, functional design, technical design, build & construction, deployment, and operations is the result of this phase.

DESIGN

The contents of the test plan will include the test cases and the testing techniques required for the acceptance of the data integration system. Test cases should be designed based on the expected results for each requirement. Traceability of the test cases to each requirement will ensure a complete design and that no requirements go untested.

Test Case design should also include the approach of testing for unexpected results such as atypical volumes or scenarios that are possible. The design should consider the inputs, dependencies, and evaluation approaches.

EXECUTION

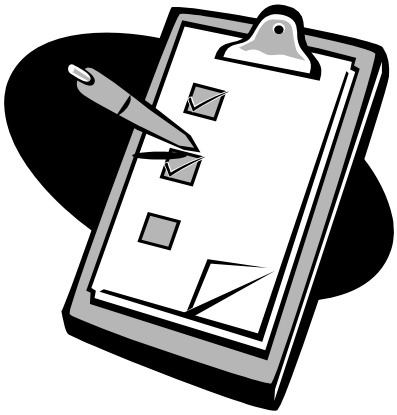
The execution of the test plan takes place as soon as the development effort begins. Tests occur throughout the development lifecycle and typically the project phases are not closed out until the test plan for that phase is completed. Due to the iterative nature of data integration development, testing can occur in various phases. Coordination and tracking of the test plan results is an activity that occurs in the execution phase.

REVIEW

Reviews occur at various stages through the testing lifecycle. A review will occur to accept the initial plan and continuously throughout the execution of the test cases. Reviews of testing results will determine the quality and acceptability of the system.

Testing Practices

Test Cases



- Identification
- Specification of Inputs
- Specification of pre-conditions
- Specification of expected outcomes

Testing Practices

Test Cases

TEST CASES

IEEE (Institute of Electrical and Electronics Engineers) defines a test case as a "set of inputs, execution pre-conditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement." A set of test cases addressing all of the requirements of a system is called a test suite.

Testing of DI systems can be complex. It is not possible to test everything prior to deployment, however it is possible to try and be thorough. A test case is a small but fundamentally important building block in testing of software and systems. Test cases bring structure to the process of testing. A complete test case includes:

- **Identification** - designating the component or logic path to be tested.
- **Specification of inputs** to be used for testing – data, runtime parameters, etc.
- **Specification of pre-conditions** – data and system states that must exist before the test is executed.
- **Specification of expected outcomes** from the test – data and system states that should exist after the test is executed.

Test cases are reusable objects. When managed as a collection of test suites, they are useful not only for initial testing of a system but for re-testing when changes occur and when regression testing is needed. A library of test cases accelerates the testing process, reduces the labor required for regression testing, and helps to ensure completeness of testing.

Testing Techniques

Stress Testing

review/walkthrough
black/white box test
inspect/profile
observe
stress test

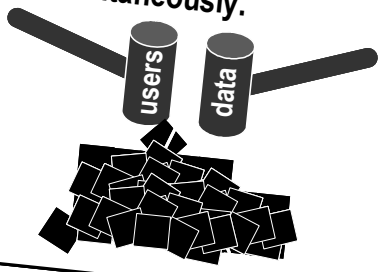
Purpose:

To discover weaknesses that will appear when the product is under stress from high volume of users, transactions, or data –

To identify weaknesses, not to fix them.

Process:

Exercise the product with high data volume, with high user and transaction volume, and with both simultaneously.



Results:

- Known threshold points at which the product will perform unsatisfactorily due to stress.
- Known threshold points at which the product will become unstable due to stress.
- Known threshold points at which the product may fail due to stress.

Testing Techniques

Stress Testing

UNDER PRESSURE

Stress testing, also known as load testing, is performed to determine the threshold points at which a product will perform unsatisfactorily, will become unstable, or will fail due to stress. Exercising a system with unusually high volumes of data, exceptionally large numbers of simultaneous users, and uncharacteristically high transaction volumes can expose vulnerabilities not found under conditions of normal use.

Stress testing is important to identify the threshold points. Once fail-safe levels are known they can be evaluated against anticipated peak conditions and against service level agreements to determine whether they represent quality defects in the products being tested.

Testing an ETL process to determine thresholds for performance and the amount of data that can be processed is an example of stress testing.

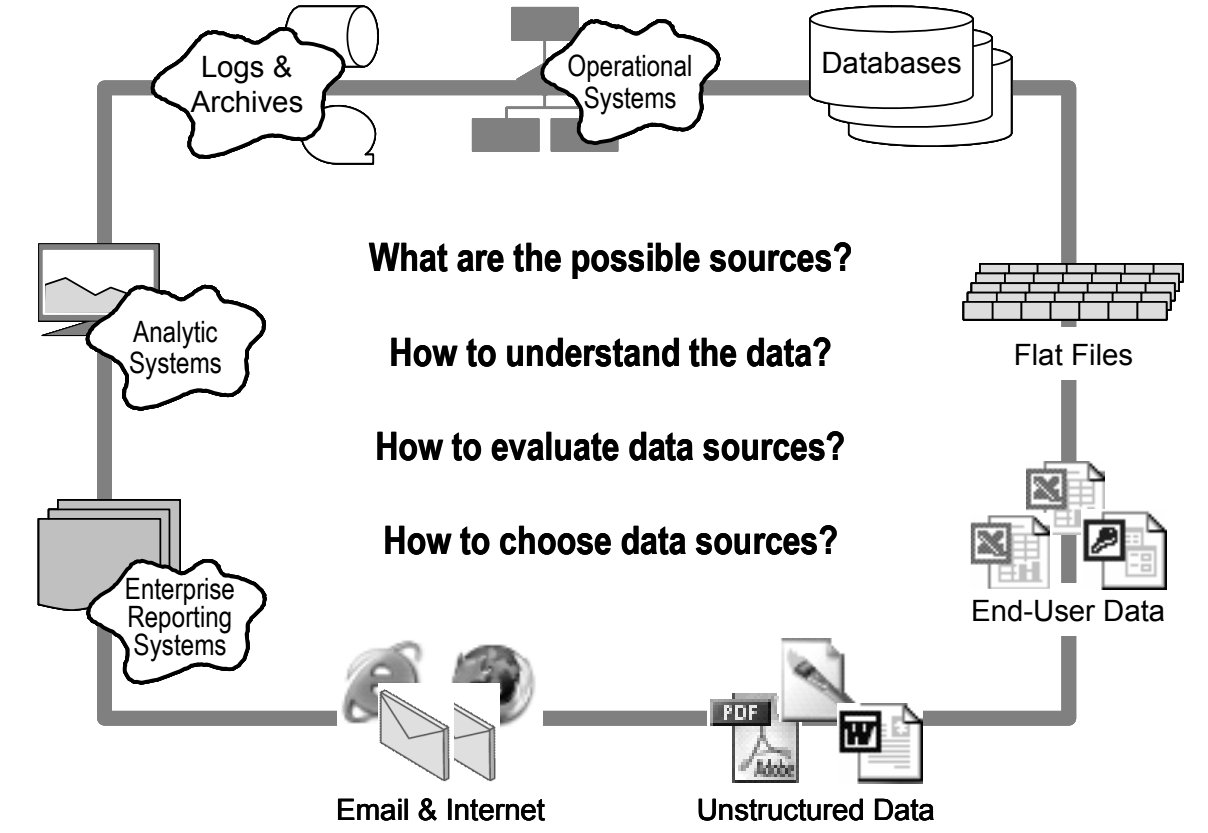
Module 4

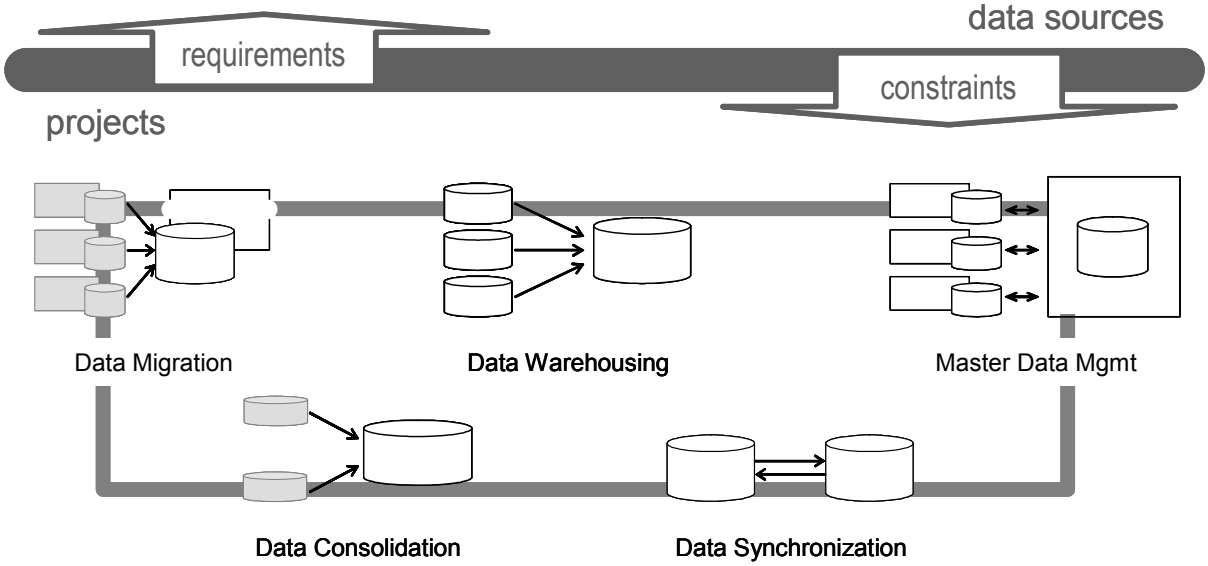
Testing through the Life Cycle

This page intentionally left blank.

Requirements Testing

Source Data Requirements





Requirements Testing

Source Data Requirements

SOURCE DATA NEEDS AND CONSTRAINTS

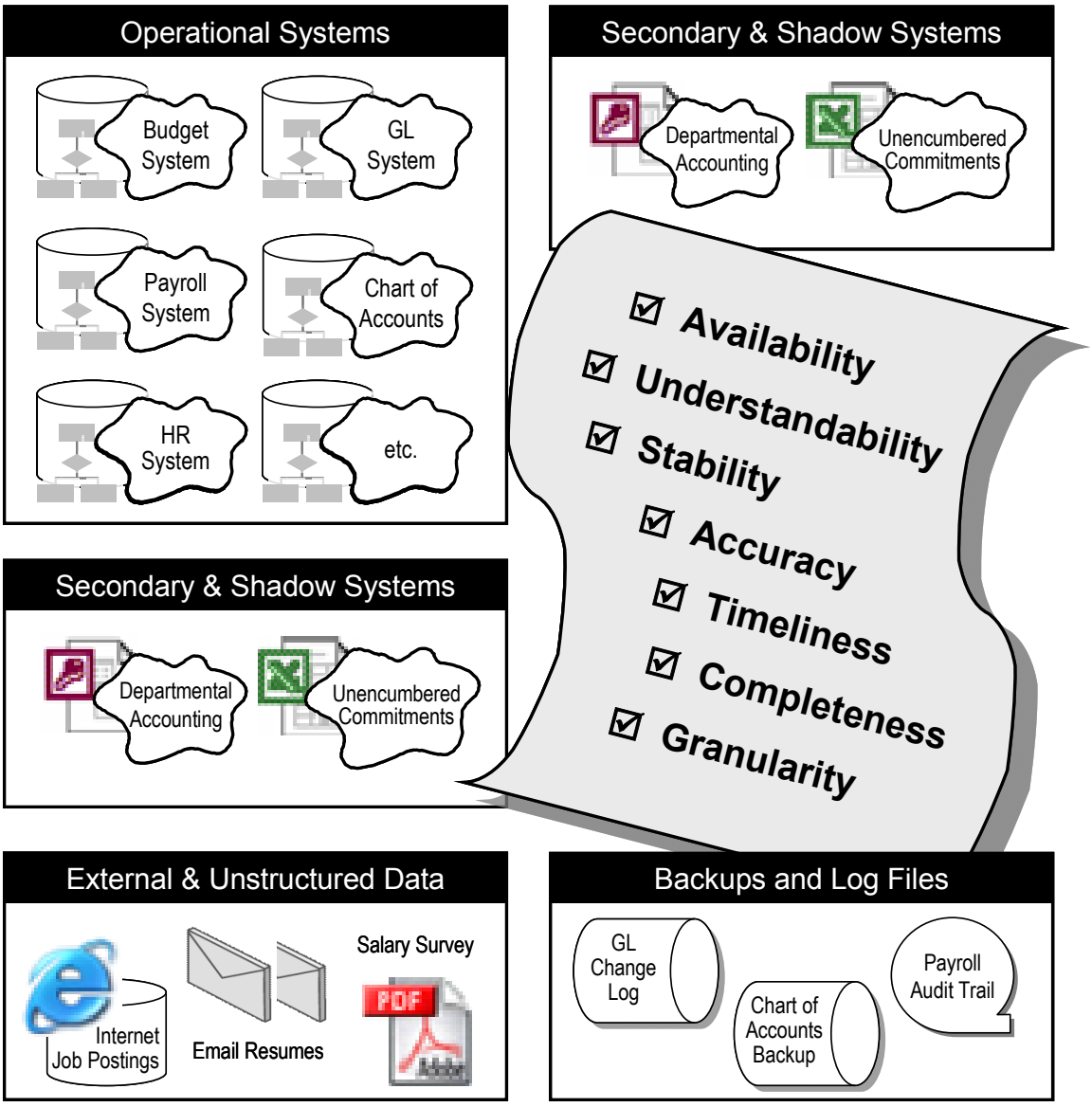
Source data requirements exist in two forms:

- The obvious requirements are those that define which data is needed to satisfy the objectives of the integration project – for example, the specific data that is needed to populate a data mart.

- Less obvious but often more challenging are the requirements (often in the form of constraints) that are imposed on the project by the nature of the source data. These requirements can only be known by thorough analysis and understanding of every potential data source. Understanding the data is difficult and time-consuming. Failing to understand the data brings even greater challenges in the form of mid-project surprises, system errors, and failed projects.

Requirements Testing

Source Data Requirements



Requirements Testing

Source Data Requirements

IDENTIFY THE CRITERIA

To identify the best source system candidate, the following qualifying criteria should be considered. When establishing the testing criteria, the dimensions of availability, understandability, stability, accuracy, timeliness, completeness, and granularity should be evaluated within the context of the user requirements.

- Is the data available to meet the requirements?
- How stable is the source system? Will the source system be too volatile to meet the requirements?
- Does the source system provide the granularity to meet the requirements?
- Does the timeliness of the source system meet the latency requirements of the users?

OTHER CRITERIA – POINT OF ENTRY AND SYSTEM OF RECORD

Point of entry, the first place that a data element is recorded anywhere in the business, is a consideration when selecting source data, and is particularly significant for data warehousing. Capturing warehouse source data at the initial point of entry improves timeliness of data by removing the time lag associated with downstream processes.

Another important criterion for data sourcing is how the business community regards a data source. If the business identifies a system as the system of record, then it certainly should be considered as a probable warehouse data source.

Requirements Testing

Source Data Requirements

Qualifying Criteria	Assessment Questions
Availability	How available and accessible is the data? Are there technical obstacles to access? Or ownership and access authority issues?
Understandability	How easily understood is the data? Is it well documented? Does someone in the organization have depth of knowledge? Who works regularly with this data?
Stability	How frequently do data structures change? What is the history of change for the data? What is the expected life span of the potential data source?
Accuracy	How reliable is the data? Do the business people who work with the data trust it?
Timeliness	When and how often is the data updated? How current is the data? How much history is available? How available is it for extraction?
Completeness	Does the scope of data correspond to the scope of the data warehouse? Is any data missing?
Granularity	Is the source the lowest available grain (most detailed level) for this data?

Requirements Testing

Source Data Requirements

WHAT TO TEST – COMPLETENESS AND CORRECTNESS

Identifying the criterion is the first step in testing the source data requirements. Once the criterion has been identified, the focus of the testing should be on completeness and correctness of the requirements. Using the testing techniques of walkthrough and profiling will work well when ensuring the completeness and correctness of the requirements.

WALKTHROUGH

Use the walkthrough to examine the completeness of the requirements related to the dimensions of availability, understandability, stability, accuracy, timeliness, and granularity. Additionally, a walkthrough is most effective when the traceability of requirements exists. Traceability provides the ability to link requirements throughout the data integration lifecycle.

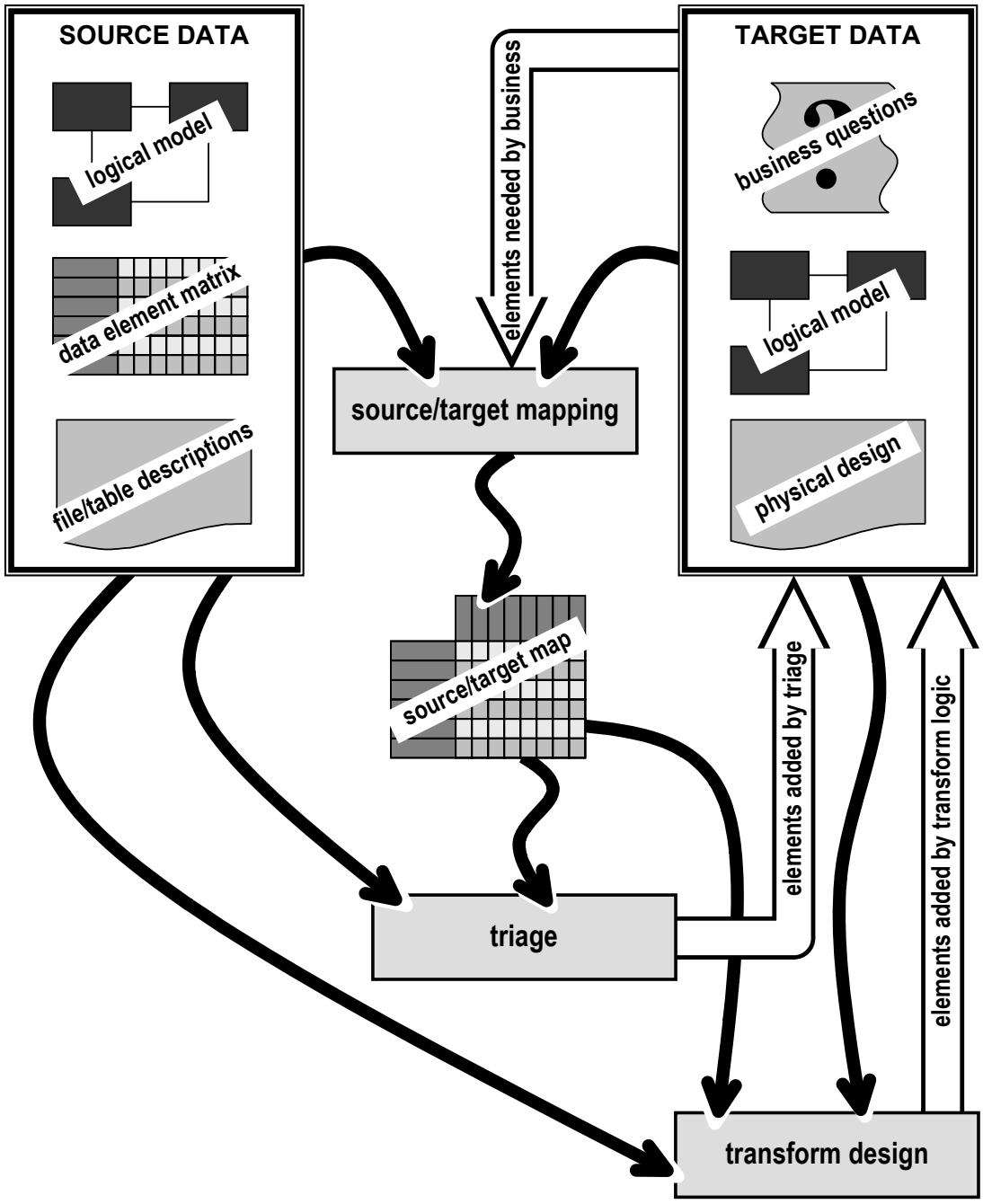
SOURCE DATA ANALYSIS AND PROFILING

Examination and profiling of source data is essential to understanding the characteristics of the data sources. This knowledge is a must for design and specification of ETL processing. Data profiling can assist in confirming source data meets requirements and if any issues exist. Profiling ensures the correctness of the data by examining the areas of availability, understandability, stability, accuracy, timeliness, and granularity. Understanding the basic structure and knowing the definitions and business rules is a good start, but it isn't enough to use a data source with confidence. Hidden structures, unexpected uses, and quality issues can be known only by looking at data contents.

The primary focus of profiling the source data is to ensure the source data has the qualities required to meet the objectives of the data integration project.

Requirements Testing

Target Data Requirements



Requirements Testing

Target Data Requirements

SOURCE AND TARGET DATA ASSOCIATIONS

Target data requirements will be based on the relationship between the target and the source data. Additionally, target requirements will be directly related to how the data is going to be used by the end user community and in what form. Target data requirements are typically created through a mapping exercise where source data is analyzed at the following levels:

- Mapping *entities* to understand the business associations
- Mapping *tables and files* to understand associations among data stores
- Mapping *columns and fields* to understand associations at the attribute level

Once the source data is analyzed, it will be clear as to what the requirements are to transform, cleanse, and load the data into the target and at what level of granularity. Transformation and cleansing requirements will be addressed later in the course.

WHAT TO TEST

The focus of the target requirements is to identify the full set of data elements. It will be important to identify:

- Which data elements are required to meet the objectives of the data integration project?
- Which data elements are needed by data transformation activities?
- Which data elements provide added value?
- How do data elements need to be formatted and presented for use?

All the data elements that may be needed for the data integration project may not be known until the mapping exercise is completed. Testing the target requirements will focus on having the full set of data elements to meet the objectives of the project, support transformation and business rule application, and identifying additional elements that provide value. The approach of "touch it – take it", sometimes known as "triage" is an approach that prescribes that all data acquired from a source is captured and stored in the data warehouse environment. Target requirements will also be impacted by the type of data integration project. For example, a data consolidation project may have the target requirements dictated by the target system.

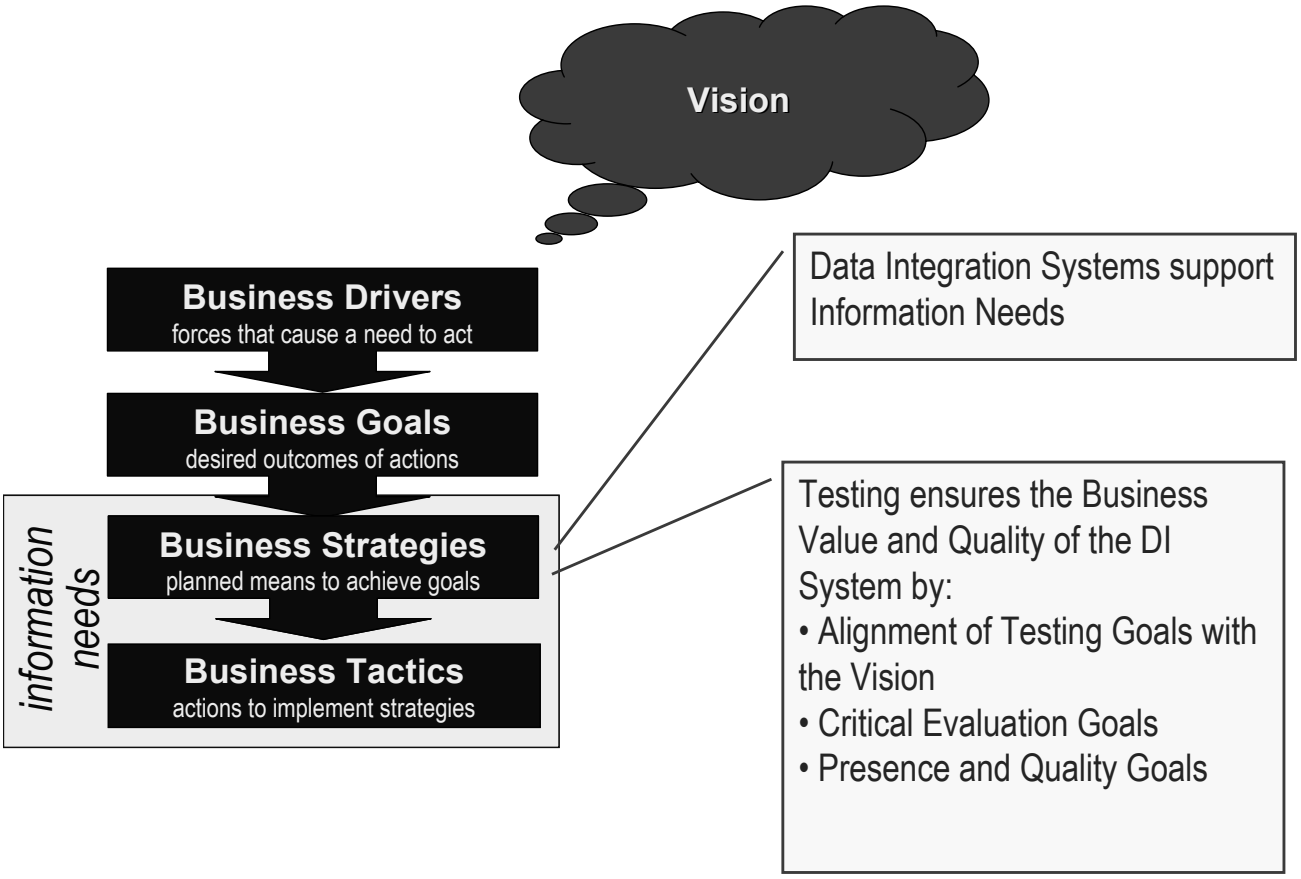
Module 5

Test Planning and Execution

This page intentionally left blank.

Test Planning

Testing Goals and Purpose



Test Planning

Testing Goals and Purpose

DEFINE THE GOALS

Effective testing begins with understanding the goals of testing – being clear about what the testing is to accomplish. The definition of testing states two distinct kinds of goals that are possible:

- **Critical Evaluation Goals** – The purpose of critical evaluation testing is to find defects; to prove that the components being tested do not work properly.

- **Presence and Quality Goals** – The purpose of presence and quality testing is to demonstrate desired qualities; to prove that the components being tested do satisfy pre-defined quality criteria.

These two goals require distinctly different attitudes from testers. To perform critical evaluation testing with the mindset of proving that the system does work will result in ineffective testing. Similarly, presence and quality testing performed to seek out failure is ineffective.

Effective testing requires the right people, with the right attitude, pursuing the right goals.

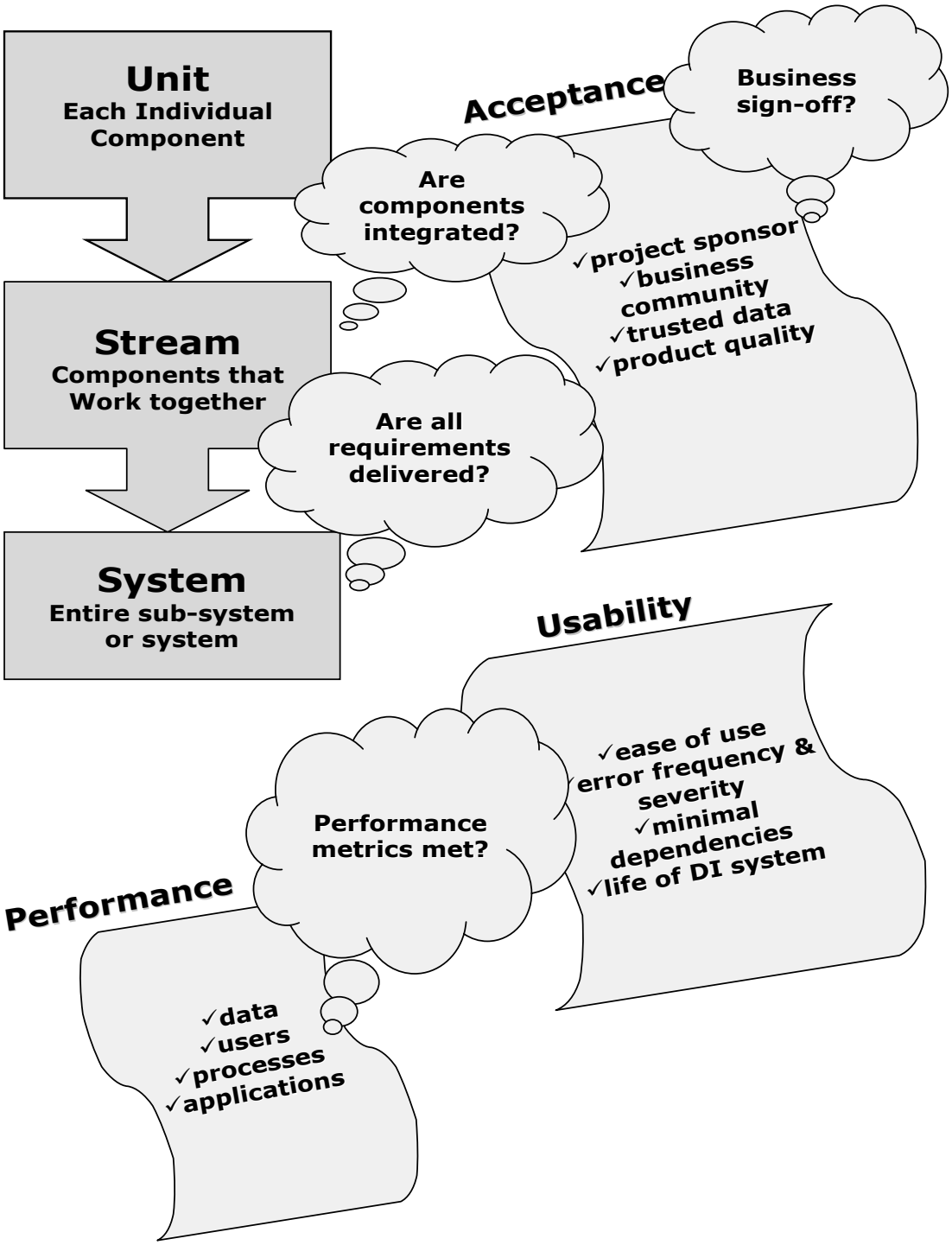
ALIGNING GOALS WITH THE VISION

Testing goals are created with an understanding of the critical evaluation goals and presence and quality goals listed above; however the specific goals for an effort will be determined by the program vision and the project requirements. Aligning the testing goals with the vision and requirements ensures that business sponsor and end user expectations are met and that the effort is supporting business value.

Data Integration systems have a critical role in supplying the information needs of an organization. Information needs directly support business strategies and tactics that, in turn, support an organization's vision.

Test Planning

Testing Metrics and Measures



Test Planning

Testing Metrics and Measures

DEMONSTRATING RESULTS

A challenge with testing is determining when enough testing has taken place or when to stop. Establishing testing metrics and measures assists in managing expectations and the testing cycle. Typical testing metrics and measures focus on two areas:

- Measuring success of a test case or test phase. The success of a test case can confirm that a requirement has been delivered.
- Establishing the exit criteria of the testing phase.

How are metrics and measures established? Most metrics are created to establish if the requirement has been delivered. Here are some examples of testing metrics and measures:

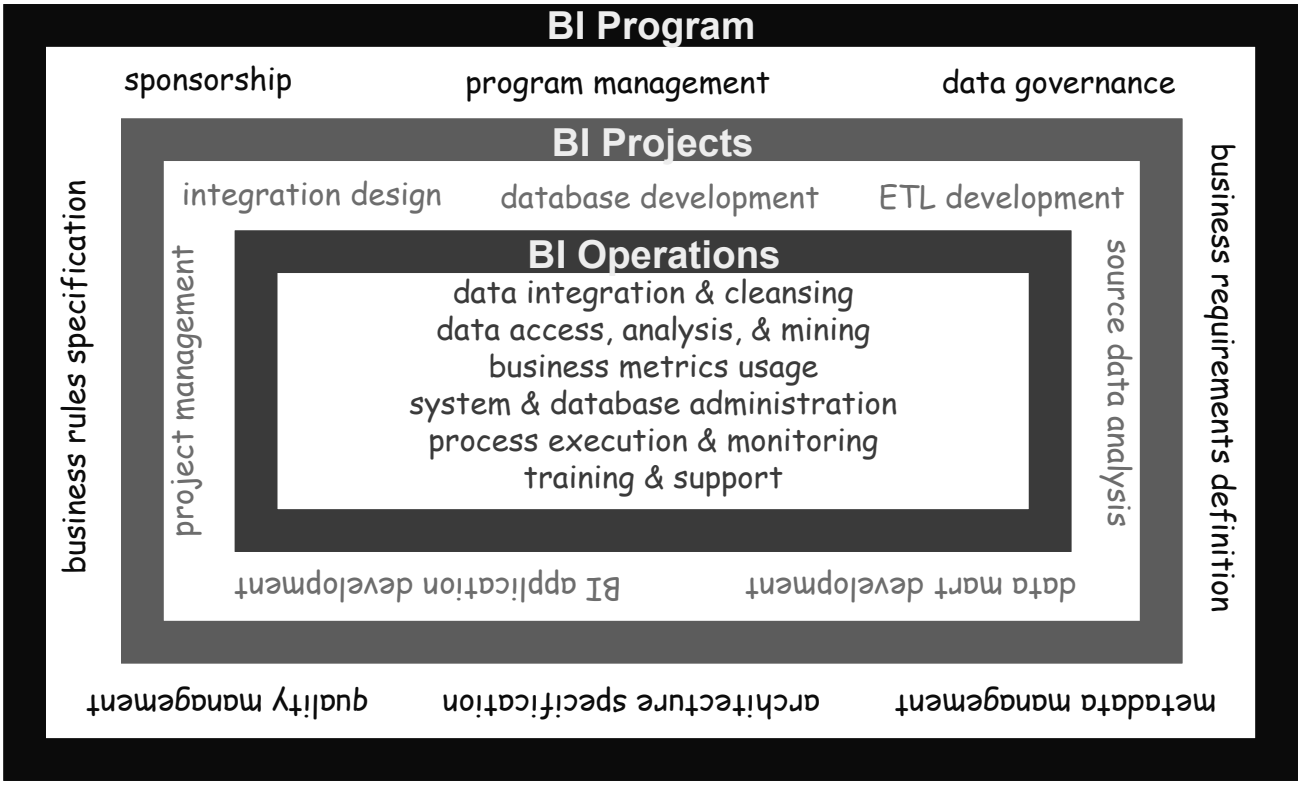
- Requirements metrics - an example of a performance requirement metric could be that the EAI process handles 300 messages an hour. Establishing requirement metrics is a good practice to ensure success for each deliverable.
- Time metrics – scheduled time for the testing cycle has passed. This is not a strong metric to indicate the overall quality of the system or the number of existing defects. Time metrics could also include time elapsed for a test case.
- Accepted number of defects - this metric would need to take into account the impact of the defects versus a number. One defect could have a larger impact on system quality than 3 minor ones.
- Formal tests are executed with no defects – all required tests have been executed with no discernable defects.
- Combination of the metrics above.

Establishing metrics and measures in the test planning phase can help quantify the overall success of the effort. Metrics should be:

- Measurable – The purpose of a metric is to assist in understanding how to apply management methods
- Independent – Metrics should be independent of human influence. Metrics should only be influenced by the event that is being measured.
- Accountable – Keep an audit trail on how the metric was established. This could include keeping test data output that was used to calculate the metric.
- Precise - Accurate enough to help in the testing results evaluation.

Test Planning

Testing Roles and Responsibilities



Test Planning

Testing Roles and Responsibilities

WHO IS INVOLVED IN TESTING?

The first thing that comes to mind when undertaking test planning is that testing will be owned by a testing organization. The testing organization works with the project team to plan, design, and execute the test plan. While this perspective may be true to an extent, testing is the responsibility of the entire team with the primary goal of delivering a high quality system. The goal of the data integration system is to provide business value. Without a quality system, there is no business value.

Testing finds defects, but does nothing to correct them. Imagine the testing cycle finding defect after defect with no focus on the overall quality of the system. What is the end result without a focus on quality? A continuous testing cycle. A development cycle that has no clear finish. Frustrated end users due to unmet expectations. Everyone must take responsibility for quality.

Quality assurance needs to be considered in each phase of the development cycle. With this in mind, testing should discover minor defects and not be the means to determine system quality.

EXAMPLES OF ROLES AND RESPONSIBILITIES

Roles and responsibilities for testing may vary across organizations. Some examples of project roles and responsibilities may include:

- Project Manager – Project testing and quality
- Data Analyst – Requirements testing and acceptance
- Database Designer – Logical and physical model testing and acceptance
- ETL Developer – Unit test case creation and execution
- Test Manager – Test Plan design and execution
- Test Analyst – Stream and system test
- Business Sponsor – Acceptance testing completion
- End User – Acceptance test cases and plans

The diagram on the prior page provides a view of different roles at different levels in a BI organization.