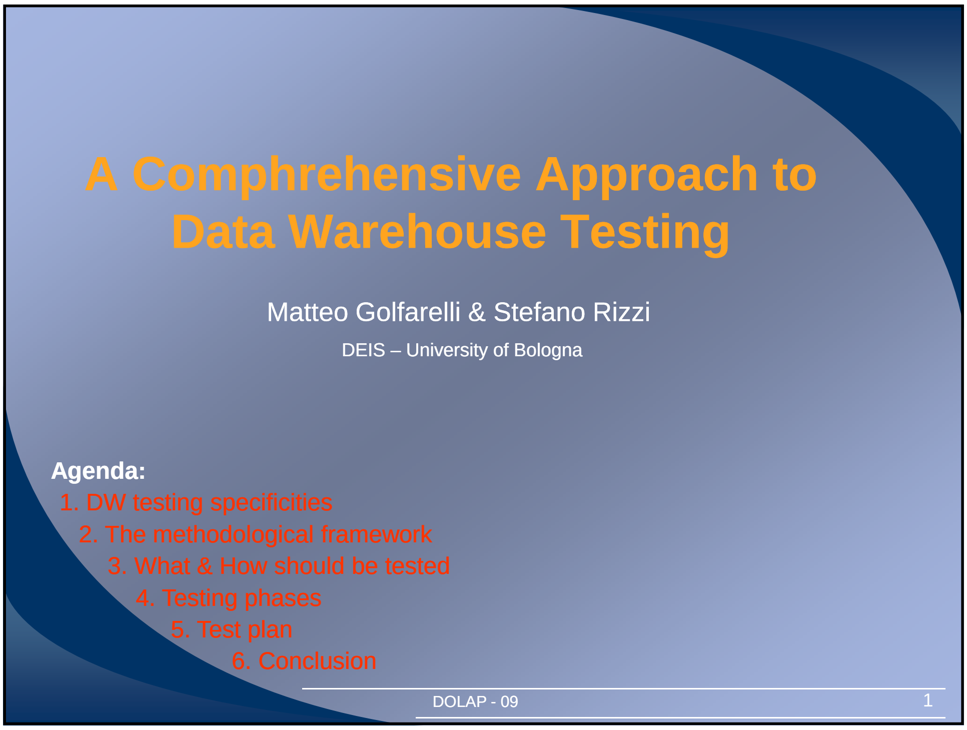


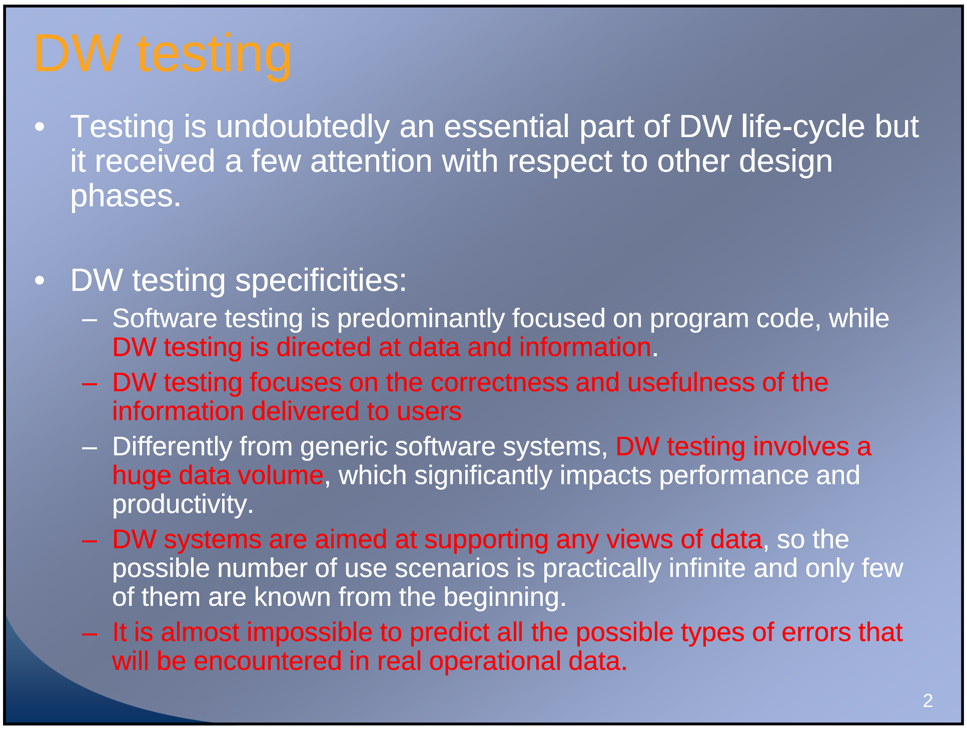
A Comphrehensive Approach to Data Warehouse Testing

Matteo Golfarelli & Stefano Rizzi

DEIS – University of Bologna

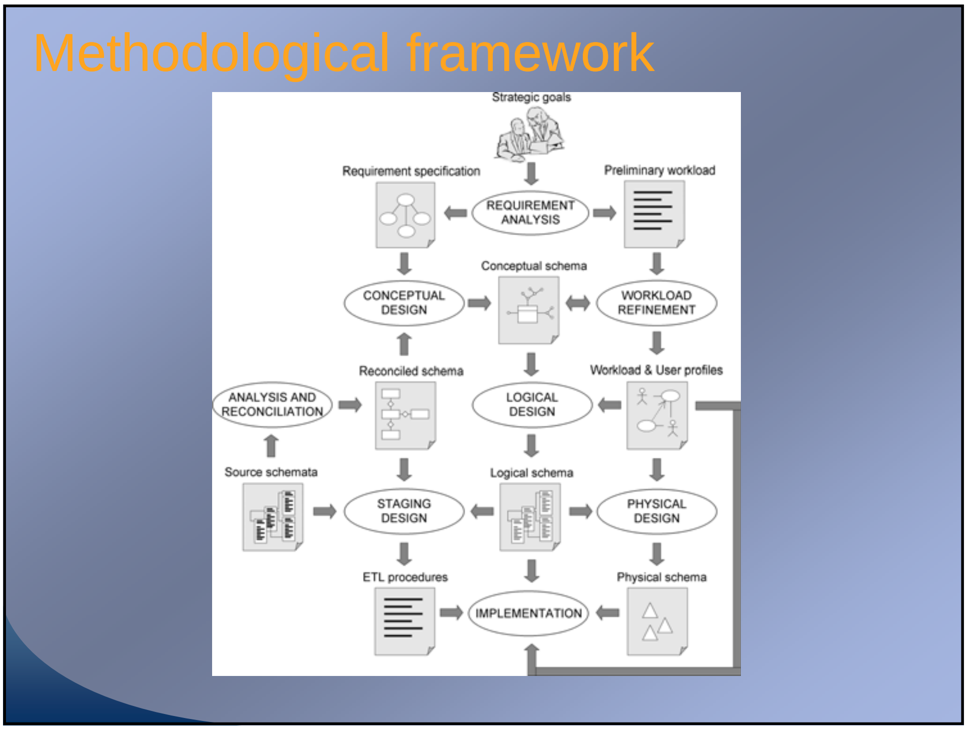
Agenda:

1. DW testing specificities
2. The methodological framework
3. What & How should be tested
4. Testing phases
5. Test plan
6. Conclusion

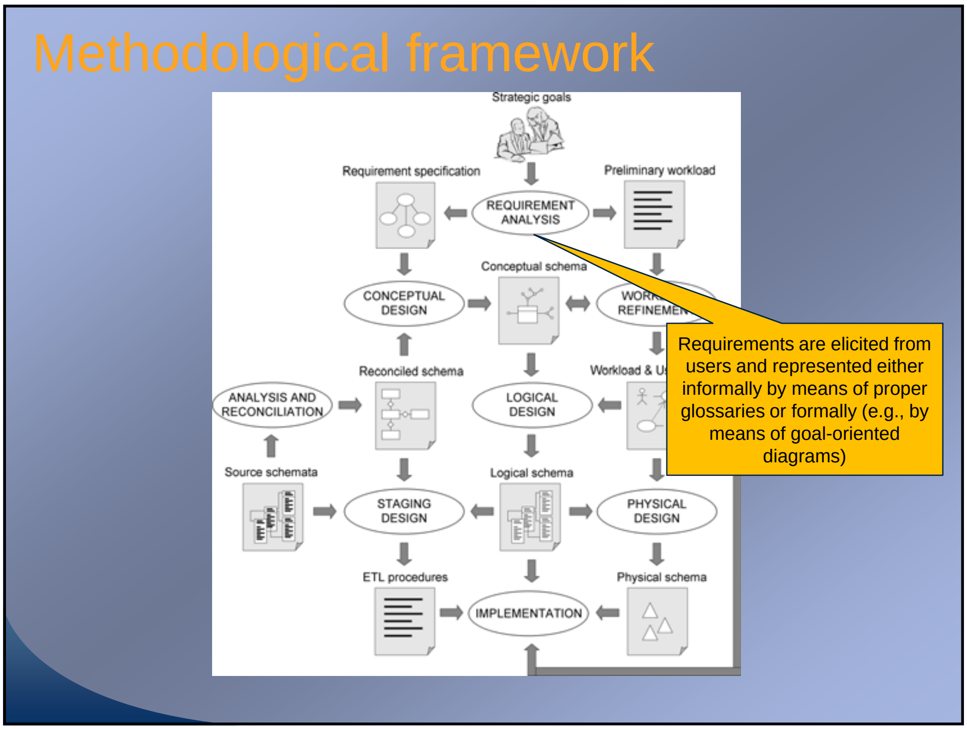


DW testing

- Testing is undoubtedly an essential part of DW life-cycle but it received a few attention with respect to other design phases.
- DW testing specificities:
  - Software testing is predominantly focused on program code, while DW testing is directed at data and information.
  - DW testing focuses on the correctness and usefulness of the information delivered to users
  - Differently from generic software systems, DW testing involves a huge data volume, which significantly impacts performance and productivity.
  - DW systems are aimed at supporting any views of data, so the possible number of use scenarios is practically infinite and only few of them are known from the beginning.
  - It is almost impossible to predict all the possible types of errors that will be encountered in real operational data.



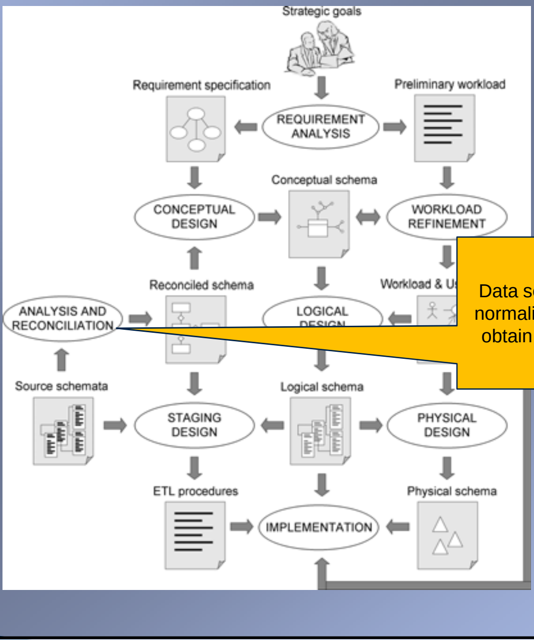
Methodological framework



Methodological framework

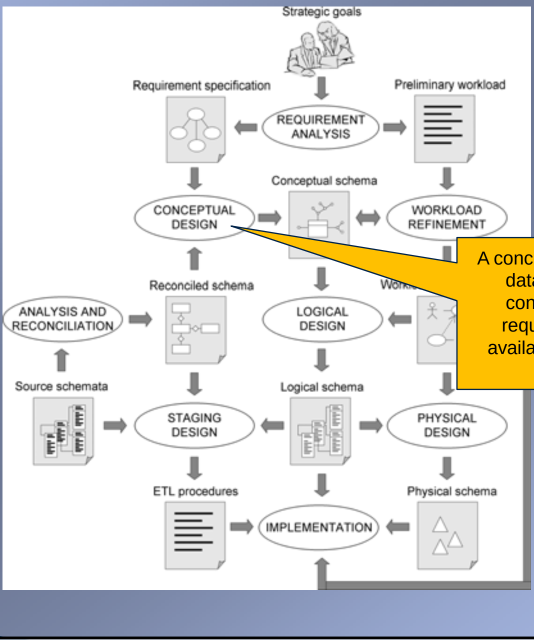
Requirements are elicited from users and represented either informally by means of proper glossaries or formally (e.g., by means of goal-oriented diagrams)

Methodological framework

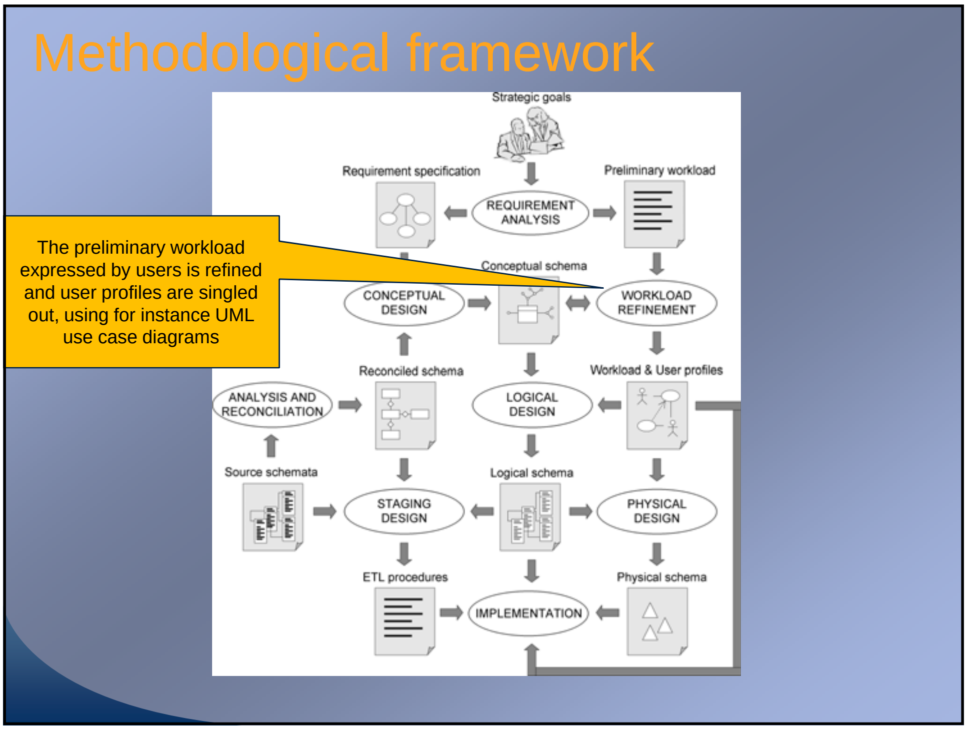


Data sources are inspected, normalized, and integrated to obtain a reconciled schema

Methodological framework



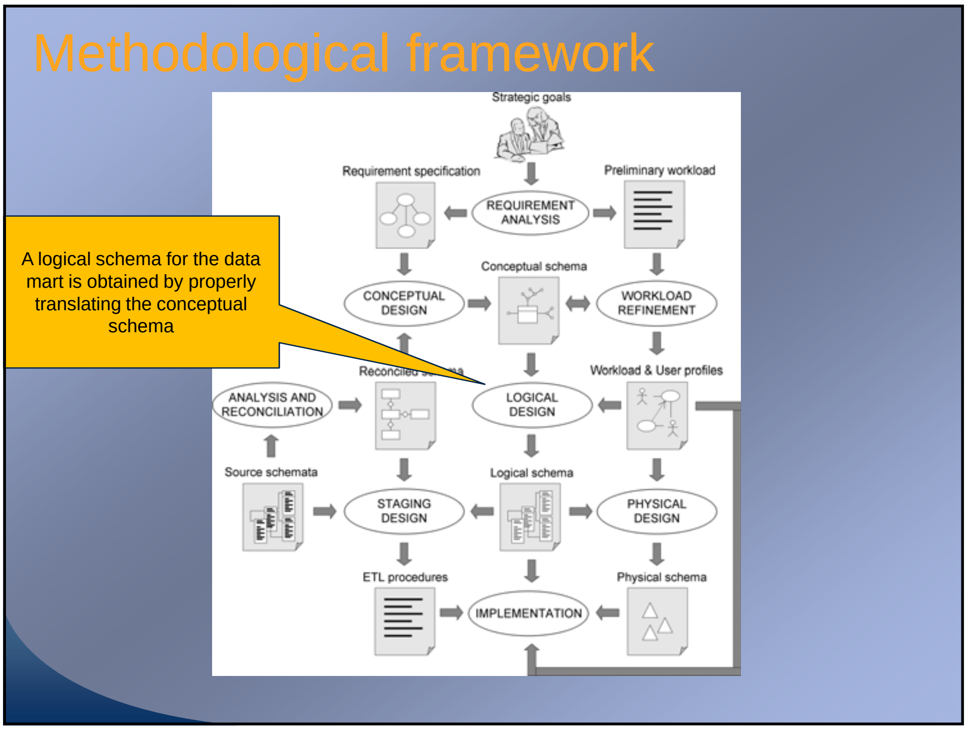
A conceptual schema for the data mart is designed considering both user requirements and data available in the reconciled schema



Methodological framework

Strategic goals
Requirement specification
REQUIREMENT ANALYSIS
Preliminary workload
Conceptual schema
CONCEPTUAL DESIGN
WORKLOAD REFINEMENT
Reconciled schema
Workload & User profiles
ANALYSIS AND RECONCILIATION
LOGICAL DESIGN
Source schemata
Logical schema
STAGING DESIGN
PHYSICAL DESIGN
ETL procedures
Physical schema
IMPLEMENTATION

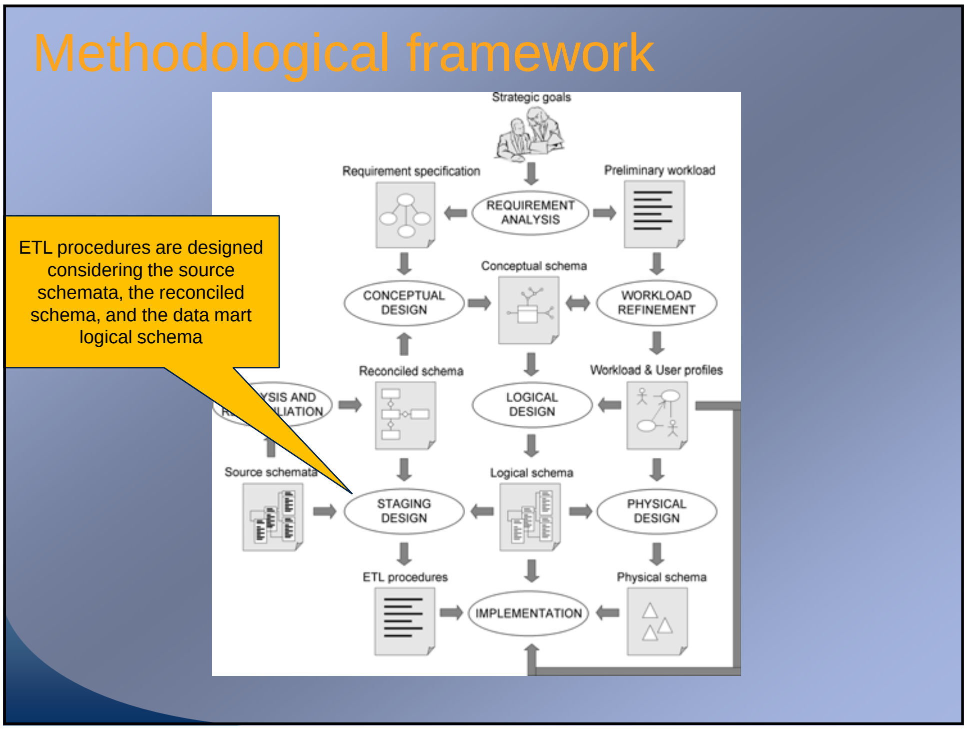
The preliminary workload expressed by users is refined and user profiles are singled out, using for instance UML use case diagrams



Methodological framework

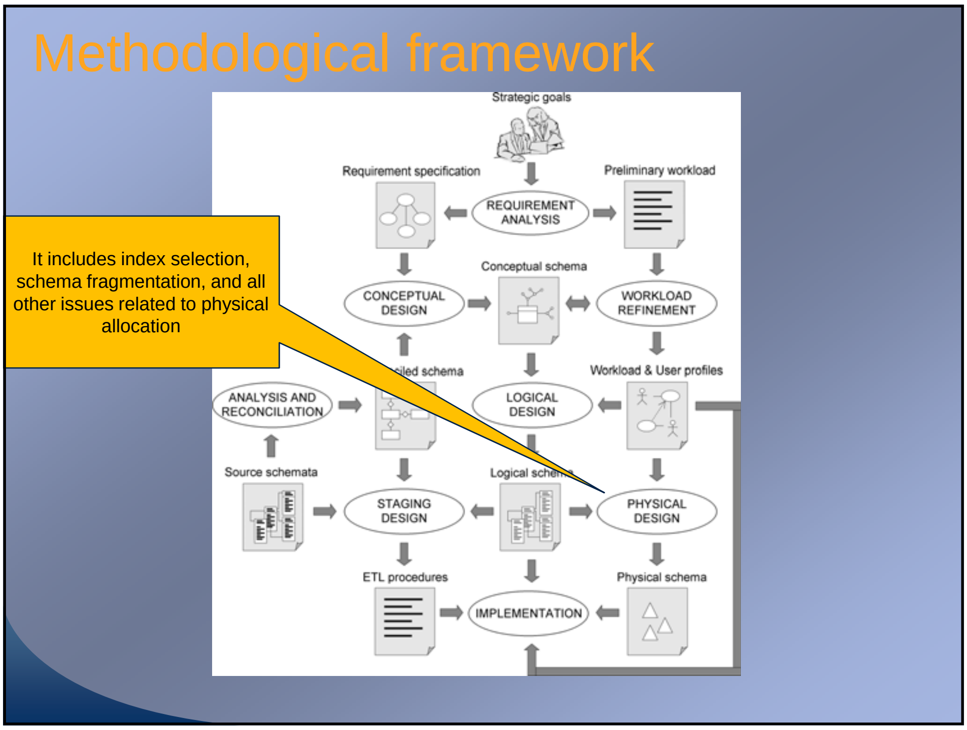
Strategic goals
Requirement specification
REQUIREMENT ANALYSIS
Preliminary workload
Conceptual schema
CONCEPTUAL DESIGN
WORKLOAD REFINEMENT
Reconciled schema
Workload & User profiles
ANALYSIS AND RECONCILIATION
LOGICAL DESIGN
Source schemata
Logical schema
STAGING DESIGN
PHYSICAL DESIGN
ETL procedures
Physical schema
IMPLEMENTATION

A logical schema for the data mart is obtained by properly translating the conceptual schema



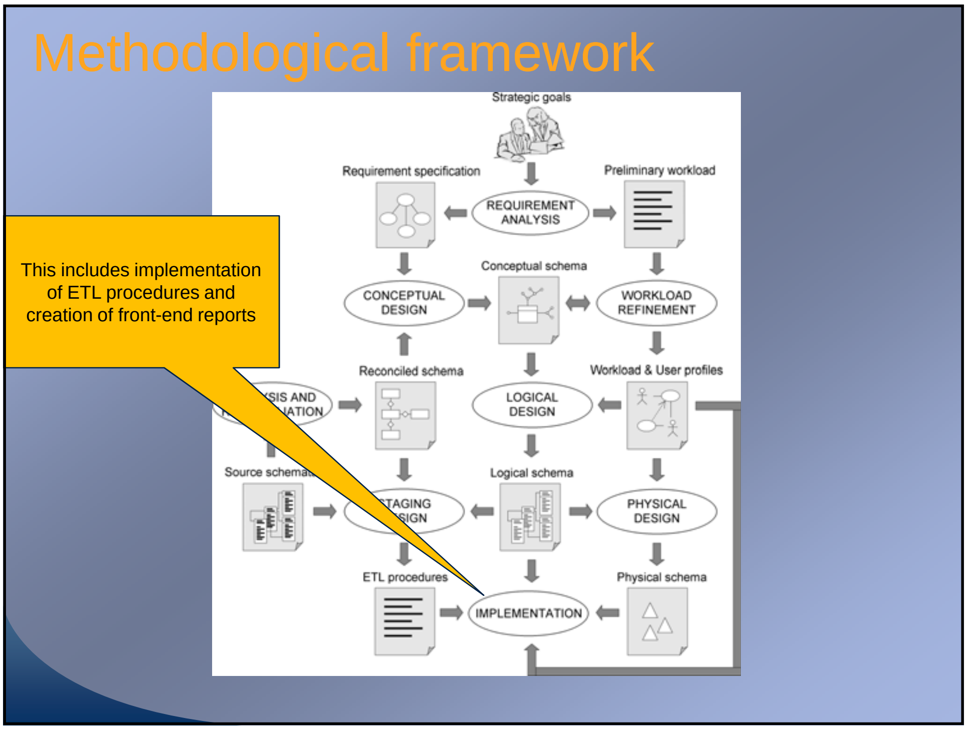
Methodological framework

ETL procedures are designed considering the source schemata, the reconciled schema, and the data mart logical schema



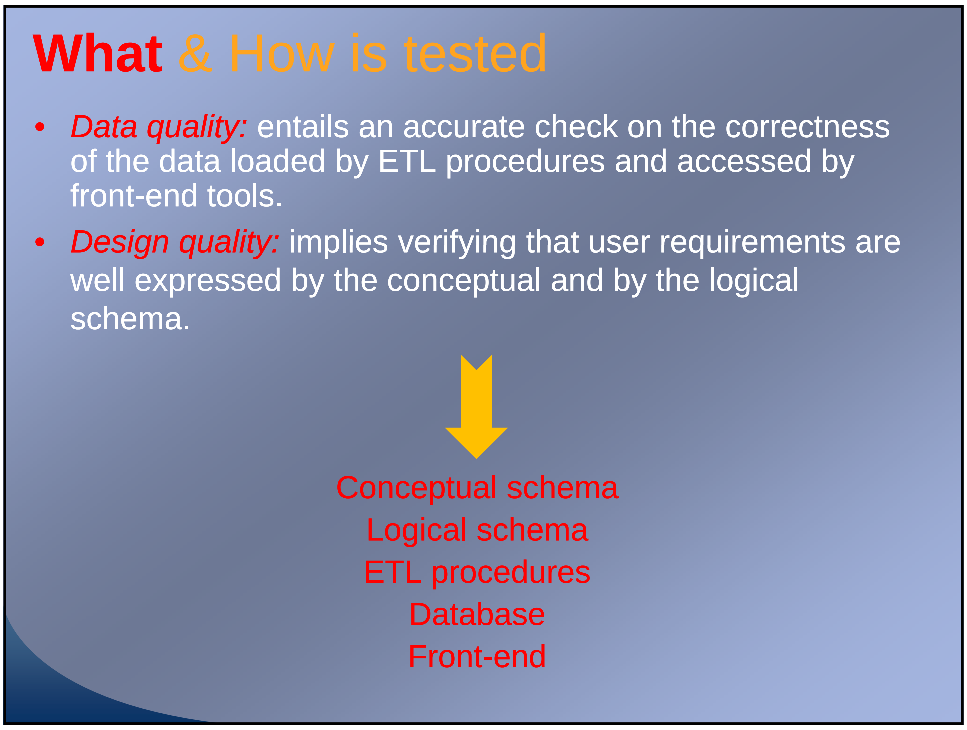
Methodological framework

It includes index selection, schema fragmentation, and all other issues related to physical allocation



Methodological framework

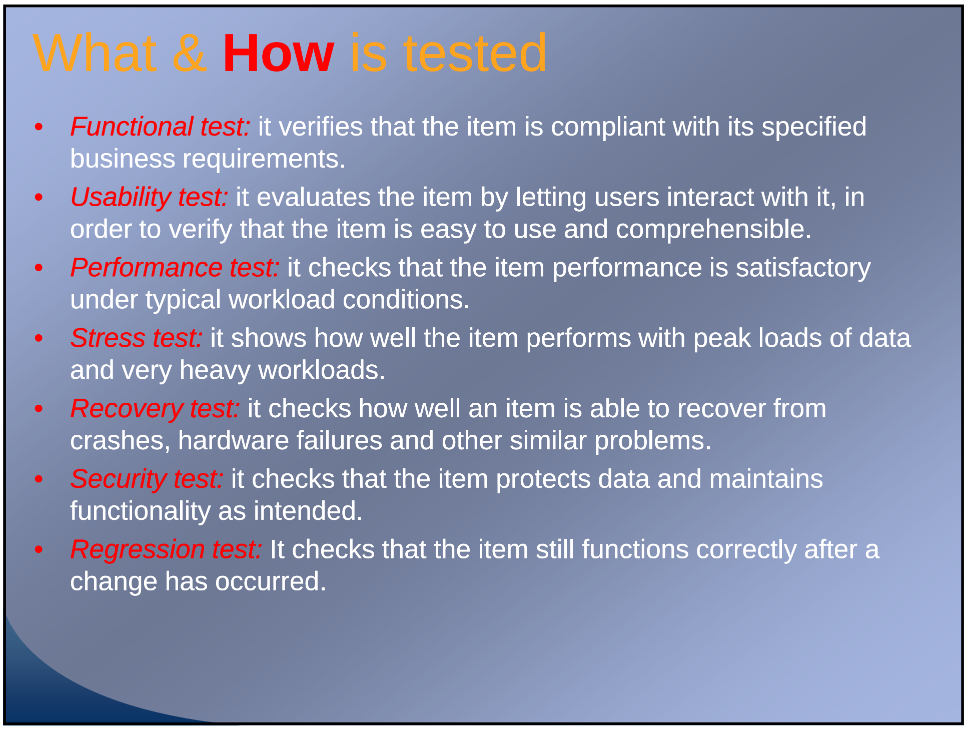
This includes implementation of ETL procedures and creation of front-end reports



What & How is tested

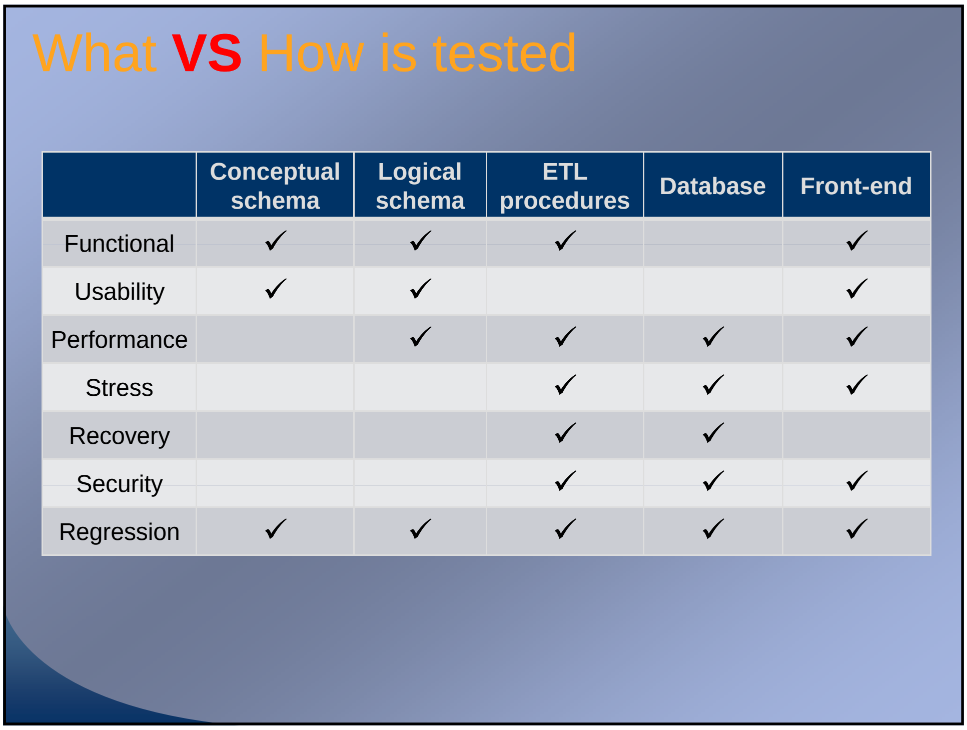
- *Data quality:* entails an accurate check on the correctness of the data loaded by ETL procedures and accessed by front-end tools.
- *Design quality:* implies verifying that user requirements are well expressed by the conceptual and by the logical schema.

Conceptual schema
Logical schema
ETL procedures
Database
Front-end



What & How is tested

- *Functional test:* it verifies that the item is compliant with its specified business requirements.
- *Usability test:* it evaluates the item by letting users interact with it, in order to verify that the item is easy to use and comprehensible.
- *Performance test:* it checks that the item performance is satisfactory under typical workload conditions.
- *Stress test:* it shows how well the item performs with peak loads of data and very heavy workloads.
- *Recovery test:* it checks how well an item is able to recover from crashes, hardware failures and other similar problems.
- *Security test:* it checks that the item protects data and maintains functionality as intended.
- *Regression test:* It checks that the item still functions correctly after a change has occurred.

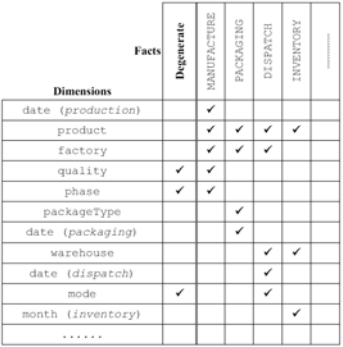


What **VS** How is tested

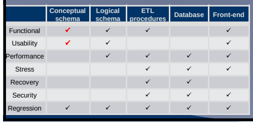
	Conceptual schema	Logical schema	ETL procedures	Database	Front-end
Functional	✓	✓	✓		✓
Usability	✓	✓			✓
Performance		✓	✓	✓	✓
Stress			✓	✓	✓
Recovery			✓	✓	
Security			✓	✓	✓
Regression	✓	✓	✓	✓	✓

Testing the Conceptual Schema

- **Functional test (Fact test):** verifies that the workload preliminarily expressed by users during requirement analysis is actually supported by the conceptual schema.
  - Can be quantitatively measured as the number of non supported queries
- **Functional test (Conformity test):** it is aimed at assessing how well conformed hierarchies have been designed. This test can be carried out by measuring the sparseness of the bus matrix
  - **Sparse bus matrix:** the designer probably failed to recognize the semantic and structural similarities between apparently different hierarchies.
  - **Dense bus matrix:** the designer probably failed to recognize the semantic and structural similarities between apparently different facts.

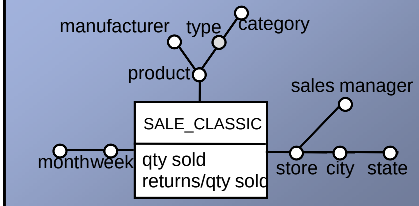


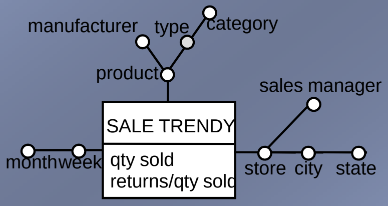
Facts Dimensions	Degenerate	MANUFACTURE	PACKAGING	DISPATCH	INVENTORY	....
date (*production*)		✓				
product		✓	✓	✓	✓	
factory		✓	✓	✓		
quality	✓	✓				
phase	✓	✓				
packageType			✓			
date (*packaging*)			✓			
warehouse				✓	✓	
date (*dispatch*)				✓		
mode	✓			✓		
month (*inventory*)					✓	
......						



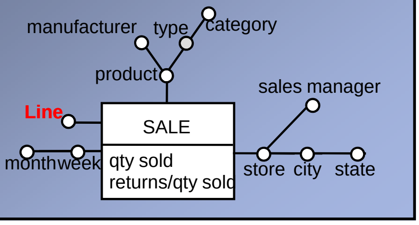
	Conceptual schema	Logical schema	ETL procedures	Database	Front-end
Functional	✓	✓	✓		✓
Usability	✓	✓			✓
Performance		✓	✓	✓	✓
Stress			✓	✓	✓
Recovery			✓	✓	
Security			✓	✓	✓
Regression	✓	✓	✓	✓	✓

Conformity Test





Facts Dimensions	SALE CLASSIC	SALE TRENDY	MANUFACTURE
Week	✓	✓	✓
Product	✓	✓	✓
Store	✓	✓	
Phase			✓
Quality			✓



Testing the Logical Schema

- **Functional test (Star test):** consists in verifying that a sample of queries in the preliminary workload can correctly be formulated in SQL on the logical schema. Priority should be given to complex queries (including many-to-many association, complex aggregations, non-standard temporal scenario)
  - Can be quantitatively measured as the number of non-supported queries
- **Usability test:** consists in verifying how easy it is for a user to understand the schema. Several metrics have been developed to evaluate this point
  - Number of dimensional attributes in a star schema
  - Number of dimension tables in a star schema
  - Number of complex constructs (e.g. many-to-many association)
  - Works by M. Piattini et al. show an high correlation between an high value of dimensional attributes/ dimension tables and the time required by an user to understand the schema.

	Conceptual schema	Logical schema	ETL procedures	Database	Front-end
Functional	✓	✓	✓		✓
Usability	✓	✓			✓
Performance		✓	✓	✓	✓
Stress			✓	✓	✓
Recovery			✓	✓	
Security			✓	✓	✓
Regression	✓	✓	✓	✓	✓

Testing ETL Procedures

- **Functional testing:** is probably the most complex and critical testing phase, because it directly affects the quality of data.
  - **Unit test:** a white-box test that each developer carries out on the units (s)he developed. They allow for breaking down the testing complexity, and they also enable more detailed reports on the project progress to be produced
  - **Integration test:** a black-box test that allows the correctness of data flows in ETL procedures to be checked
  - **Forced-error test:** are designed to force ETL procedures into error conditions aimed at verifying that the system can deal with faulty data as planned during requirement analysis.
- Since ETL is heavily code-based, most standard metrics for generic software system testing can be reused here.

	Conceptual schema	Logical schema	ETL procedures	Database	Front-end
Functional	✓	✓	✓		✓
Usability	✓	✓			✓
Performance		✓	✓	✓	✓
Stress			✓	✓	✓
Recovery			✓	✓	
Security			✓	✓	✓
Regression	✓	✓	✓	✓	✓

Testing the Front-End

- **Functional testing** of the front-end must necessarily involve the end-users, they can more easily detect abnormality in data caused by:
  - Faulty ETL procedures
  - Incorrect data aggregations or selections in front-end tools
  - Poor data quality of the source database.

- An alternative approach to front-end functional testing consists in comparing the results of OLAP analyses with those obtained by directly querying the source DBs.

- **Usability tests** are carried out measuring the number of misunderstandings of the users about the real meaning of data when analyzing the reports.

	Conceptual schema	Logical schema	ETL procedures	Database	Front-end
Functional	✓	✓	✓		✓
Usability	✓	✓			✓
Performance		✓	✓	✓	✓
Stress			✓	✓	✓
Recovery			✓	✓	
Security			✓	✓	✓
Regression	✓	✓	✓	✓	✓

Testing performances

- **Performance** should be separately tested for:
  - Database: requires the definition of a reference workload in terms of number of concurrent users, types of queries, and data volume
  - Front-end: requires the definition of a reference workload in terms of number of concurrent users, types of queries, and data volume
  - ETL prcedures: requires the definition of a reference data volume for the operational data to be loaded

- **Stress test:** simulates an extraordinary workload due to a significantly larger amount of data and queries. Variables that should be considered to stress the system are:
  - Database: number of concurrent users, types of queries and data volume
  - Front-end: number of concurrent users, types of queries, and data volume
  - ETL procedures: data volume

- The expected quality level can be expressed in terms of response time

	Conceptual schema	Logical schema	ETL procedures	Database	Front-end
Functional	✓	✓	✓		✓
Usability	✓	✓			✓
Performance		✓	✓	✓	✓
Stress			✓	✓	✓
Recovery			✓	✓	
Security			✓	✓	✓
Regression	✓	✓	✓	✓	✓

Regression test and test automation

- Regression test is aimed at making sure that any change applied to the system does not jeopardize the quality of preexisting, already tested features.
  - In regression testing it is often possible to validate test results by just checking that they are consistent with those obtained at the previous iteration.
  - Impact analysis is aimed at determining what other application objects are affected by a change in a single application object. Remarkably, some ETL tool vendors already provide some impact analysis functionalities.
  - Automating testing activities plays a basic role in reducing the costs of testing activities.
- Testing automation is practically possible for a limited number of test types:
  - Implementation-related testing activities can be automated using commercial tools (e.g. QACenter) that simulate specific workloads and analysis sessions, or that can measure an ETL process outcome.
  - Design-related testing activities require solutions capable of accessing meta-data repositories and the DBMS catalog.

	Conceptual schema	Logical schema	ETL procedures	Database	Front-end
Functional	✓	✓	✓		✓
Usability	✓	✓			✓
Performance		✓	✓	✓	✓
Stress			✓	✓	✓
Recovery			✓	✓	
Security			✓	✓	✓
Regression	✓	✓	✓	✓	✓

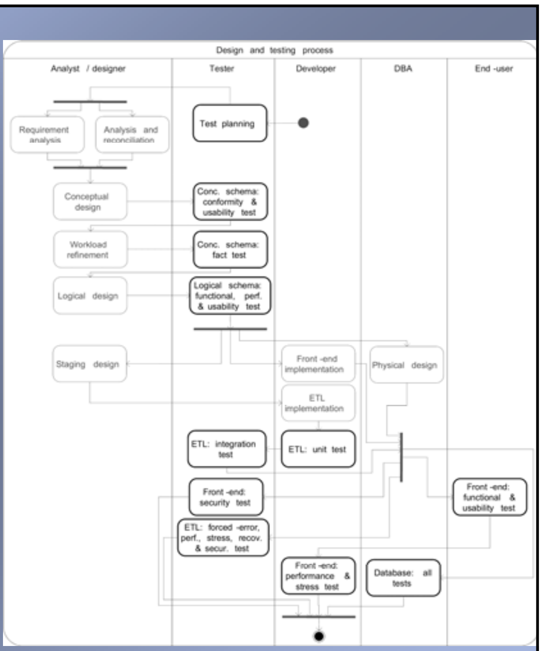
Coverage Criteria

- Measuring the coverage of tests is necessary to assess the overall system reliability
  - Requires the definition of a suitable coverage criterion
  - Coverage criteria are chosen by trading o test effectiveness and efficiency

Testing activity	Coverage criterion	Measurement	Expected coverage
Fact test	Each information need, expressed by users during requirement analysis, must be tested	Percentage of queries in the preliminary workload that are supported by the conceptual schema	Partial, depending on the extent of the preliminary workload
Conformity test	All data mart dimensions must be tested	Bus matrix sparseness	Total
Usability test of the conceptual schema	All facts, dimensions, and measures must be tested	Conceptual metrics	Total
ETL unit test	All decision points must be tested	Correct loading of the test data sets	Total
ETL forced-error test	All error types specified by users must be tested	Correct loading of the faulty data sets	Total
Front-end unit test	At least one group-by set for each attribute in the multidimensional lattice of each fact must be tested	Correct analysis result of a real data set	Total

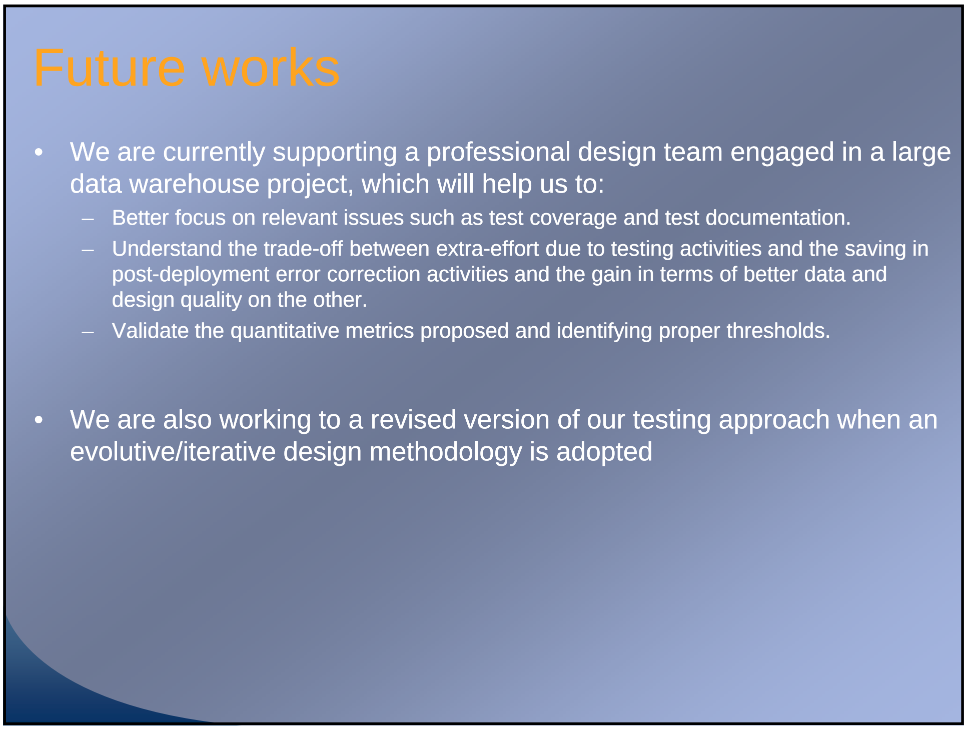
A test plan

1. **Create a test plan** describing the tests that must be performed
2. **Prepare test cases:** detailing the testing steps and their expected results, the reference databases, and a set of representative workloads.
3. **Execute tests**


Design and testing process
Analyst / designer
Tester
Developer
DBA
End -user
Requirement analysis
Analysis and reconciliation
Test planning
Conceptual design
Conc. schema: conformity & usability test
Workload refinement
Conc. schema: fact test
Logical design
Logical schema: functional, perf. & usability test
Staging design
Front -end implementation
Physical design
ETL implementation
ETL: integration test
ETL: unit test
Front -end: security test
Front -end: functional & usability test
ETL: forced -error, perf., stress, recov. & secur. test
Front -end: performance & stress test
Database: all tests

Conclusions and lesson learnt

- The chance to perform an effective test depends on the documentation completeness and accuracy in terms of collected requirements and project description.
- The test phase is part of the data warehouse life-cycle
  - The test phase should be planned and arranged at the beginning of the project
- Testing is not a one-man activity.
  - The testing team should include testers, developers, designers, database administrators, and end-users, and it should be set up during the project planning phase.
- Testing of data warehouse systems is largely based on data.
  - Accurately preparing the right data sets is one of the most critical activities to be carried out during test planning.
- While testing must come to an end someday, data quality certification is an ever lasting process.
  - The borderline between testing and certification clearly depends on how precisely requirement were stated and on the contract that regulates the project.



Future works

- We are currently supporting a professional design team engaged in a large data warehouse project, which will help us to:
  - Better focus on relevant issues such as test coverage and test documentation.
  - Understand the trade-off between extra-effort due to testing activities and the saving in post-deployment error correction activities and the gain in terms of better data and design quality on the other.
  - Validate the quantitative metrics proposed and identifying proper thresholds.

- We are also working to a revised version of our testing approach when an evolutive/iterative design methodology is adopted