Organizing Database Project Work

(Chapter 4)

"Learn the rules like a pro, so you can break them like an artist"

Pablo Picasso

Database Life Cycle

- The life cycle starts when the need is identified and continues through requirements, design, construction, deployment, maintenance, and finally, retirement.
  - Data modeling typically occurs early in the cycle because developers need the data structures before they can start coding.
  - When aggressive schedule or methodologies are used, there is enormous pressure to deliver data structures quickly.

Reality Sets In: The Project Triangle
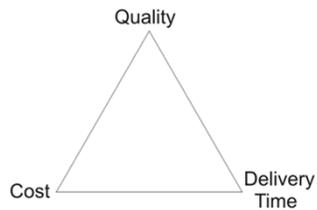
4

Real World Challenges

- Ensuring that the project plan allows for the development and proper use of high quality data models.

- Developing a series of deliverables that will culminate in a complete physical data model, and provide sufficient information for other participants in the project to carry out their work.

- Ensuring adaptability of data models and databases

- Finding the right level of participation (extremes are one person doing everything versus entire departments participating)

5

Traditional SDLC
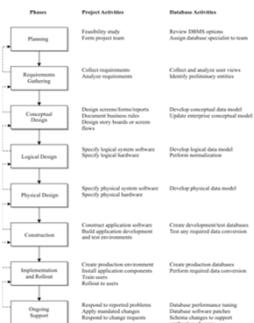
6

Planning

- Organization must determine:
  - Where they are
  - Where they want to be
  - Reasonable plan(s) to get there
- Often occurs over a longer time span than projects
- Process-driven versus data-driven approaches
- Do data(base) specialists participate?
- Who else should participate?

7

Requirements Gathering

- Complete and accurate requirements are nothing less than essential to a successful project
- Focus on *what* rather than *how*
- Beware of analysis paralysis
- From a data modeling perspective, user views are the primary interest, but business rules are also important
- Subject matter experts (SMEs) and business analysts are the "worker bees" during requirements gathering
- Consider prototyping if requirement are sketchy

8

Requirement Gathering Methods

- Conducting Interviews
- Conducting Surveys
- Observing Business Operations
- Reviewing Documentation

- What are the strengths and weaknesses of each?
- Can we think of other methods?

9

Conceptual Design

- Involves designing the externals of the application(s) and database(s)
- Refinement of user views and business rules may be required
- Application designers focus on screen flows, story boards and the like
- Consider workshops (e.g. JAD) if requirements need to be fleshed out
- Conceptual Data Model developed (a high-level ERD)
  - There is much debate over the right level of detail for a conceptual ERD

10

Logical Design

- Encompasses the bulk of the technical design of the application(s) and database(s)
- Application designers segment the application into modules and perform process modeling
- Data modeler creates a logical data model consisting of a logical ERD (normalized if OLTP), entity and attribute definitions (presented as the logical data dictionary), and documentation of constraints that will likely go into the physical design
- Consider multiple models if there are multiple viable design choices and debate as to which is best (postpone performance concerns to physical design)

11

Physical Design

- Logical design (application and database) is mapped to the implementation technology to be used
- Transformation of logical data model to physical data model is largely mechanical, including adding specifications for physical storage and performance
  - Often jointly completed by data modeler and DBA
- Data Model deliverables include:
  - Physical ERD (possibly multiple alternatives)
  - Physical Dictionary
  - SQL DDL to define the physical database

12

Construction

- Application developers code and test application components
- Data modeling is complete unless the developers discover defects or additional requirements (or the business changes the requirements)
- Data modeler must ensure that the developers understand the data structures and their intended use

13

Implementation and Rollout

- Implementation is the process of installing the components
- Rollout is the process of placing groups of business users on the new application(s) and database(s)
  - Cold Turkey cutover versus Phased Rollout
- A fallback plan is also important

14

Ongoing Support

- Handling of defect remediation and enhancement requests ("Is that a bug or a feature?")
  - Some defect fixes will require database changes
  - Most enhancements require database changes
- Patches to the DBMS and Operating System
- Performance Tuning
- Monitoring of resource consumption (CPU, disk, network, etc.)

15

Nontraditional Life Cycles

- Prototyping
  - Is the prototype intended to be deployed?
  - Either way, set expectations
- Rapid Application Development (RAD)
- Agile Methodologies

16