Adventures of a Development DBA: Iterative Development

Presented By

Michael Lynott

eTRANSITIONS™

11 Grist Mill Road
Lebanon, NJ 08833
www.*e*transitions.com
Tel: (800) 236-9414

What is Iterative Development?

- Divide an application deliverable into smaller, usable pieces then build them one-at-a-time.
- Over the years:
  - Agile Development
  - RAD (Rapid Application Development)
  - Spiral Methodology
  - RUP (Rational Unified Process)

Daily Chaos or ...?

- Traditional:
  - Fully analyze data and processes
  - Detailed technical design
  - Build

- Iterative Development:
  - Analyze a portion
  - Design & Build
  - Analyze more
  - Modify design & Build
  - Repeat until done!

"Start coding…, in the meantime, I'll go see what the users want!"

Principles of Iterative Methodologies

- Fixed-length Iterations
  - Timebox approach to scoping
  - Needs a strategy level definition to set scope
  - Number of iterations defined up-front
  - Typical length between 4 – 10 weeks
  - Vulnerable to incomplete deliverables

- Phased Deliverables
  - Length of time defined by selected scope
  - Traditional approach
  - Number of iterations unknown
  - Each iteration length varies based on scope
  - Vulnerable to "creeping featuritis"

Remember the Fast Food Fallacy

No difference plus

No difference plus

No difference =

A clear difference (eventually)

G. M. Weinberg (*Secrets of Consulting*)

Role of the Data Modeler

- Traditional:
  - Model entire business area
  - Obtain agreement on *all* attribute details
  - Use a significant portion of allocated project time

- Iterative Development
  - Model pre-defined portion of business
  - Obtain agreement on important attribute details
  - Educated guesses on lesser attribute details
  - Use predetermined portion of allocated iteration time

Role of the Development DBA

- Traditional:
  - Design physical database
  - Create tables, columns, constraints
  - Support development of packages & procedures
  - (Grumble about changes to schema after initial design!)

- Iterative Development:
  - Design physical database
  - Multiple schemas for iteration conversion
  - Plan for multiple copies of packages & procedures
  - Utilities for schema changes & test data migration

Challenges of project structure

- How many iterations?
- How to set the scope of each iteration?
- Time for each iteration?
- Deliverables of each iteration?
- Test data generation and migration
- Frequency of changes to schemas

App Developer to DBA: “Can you just change the data type of this column? We’ve got to have it today!”

Setting up the Environments

- Development
- Test
- Production
- Data Migration area?
- Challenge: Progressing development components from one environment to the next

Managing Development and Test data

- Kinds of data:
  - Reference data e.g. pick lists, classifications
  - Base data e.g. data fundamental to application e.g. customers, products
  - Migrated data from legacy systems
  - New data created by application

Managing Development and Test data

- Which categories of data to migrate from dev to test?
- Data volumes between dev and test?
- How to keep PK values consistent between dev and test?
- Migrating data from dev to test?
- Data Integration with other apps?

Staging of Development Code

- Code categories:
  - – Packages, procedures and views in database
  - – Middleware code (Websphere, 9iAS)
  - – Program code (Java, plug-ins, client-side)
- Issues:
  - – Configuration of a release from dev to test?
  - – Synchronizing changes between code categories

Staging of Reference Data

- Where is reference data initiated – production or development?
- Migrate it or share reference data tables (via synonyms)?
- Constraints on reference data or rule-based checking?
- Keep PK id’s consistent?
- Migrating constraint-dependent data – order of tables based on FKs

Code Control for Database Objects

- Managing versions of database-based code e.g. procedures, packages, views
- Shared views, procedures etc between applications
- Integration conflicts with other apps

Staging of Test Data

- Which data categories to migrate?
- Keep PK id’s consistent?
- Migrating constraint-dependent data – order of tables based on FKs
- Tools like Toad Data Manager?

Data Modeling Challenges

- Modeling with blinders
  - Strict adherence to scope of iteration
  - Accepting that change is inevitable with each iteration
  - Considering issues beyond scope results in defeating timebox

Data Modeling Challenges

- Being less pedantic about details
  - Agreement only on important attributes
  - “wing it” with lesser attributes
  - Use past experience (if you have any!)
  - Don’t sweat details like length – database features makes changes easy!

Development DBA challenges

- Multiple schemas
- Frequency of migration from one environment to next
- Responsible for migrating data and database-based code
- Synchronizing migration of program and middleware code

Project Management Challenges

- Setting scope of each iteration
- Controlling scope of current iteration
- Keeping blinders on all team members
- Keeping an eye on business value of each feature
  - Example: 3 weeks design/coding effort for services ordered on 5 orders per month – generating revenue worth $50!

Bringing Discipline to chaos

- Scope definition of each iteration is critical
  - Must understand technical architecture
  - Must appreciate complexity
  - Must do good estimates on effort per feature
  - Architecture must define incremental features
- Strict adherence to iteration structure
  - Commit to iterations and schedule
  - Keep to schedule, maintain credibility