A Disciplined Agile Approach to Data Warehousing (DW)/Business Intelligence (BI)

Are You Agile or Are You Fragile?
Scott Ambler

- Helps enterprise-class organizations around the world to improve their processes and organization structures
- Thought leader of:
  - Agile Modeling (AM) method
  - Agile Data (AD) method
- Co-creator of the Disciplined Agile (DA) framework
- Author of 20+ books
- scott [at] scottambler.com
- Advisory Board, ScaleFree

@scottwambler
We’re going to cover a lot of ground
Agenda

- Being Agile: An Agile Mindset for DW/BI
- Doing Agile: The Agile Database Technique Stack
- Parting Thoughts
An Agile Mindset for DW/BI
Work Collaboratively

NOT
Embrace Changing Requirements
Model to Understand, Test to Specify

• Models are very useful for
  – Exploring high-level ideas
  – Supporting conversations
  – Describing high-level ideas

• Automated tests are very useful for
  – Specifying in detail how something should work
  – Validating that something works as expected
Usage Drives Development

• Data-driven approaches are too narrowly focused
• How people use, or work with, the data to fulfill business goals is the true issue ➔ focus on usage, not the underlying data structures
• Work closely with your stakeholders to explore their intent

Usage requirements drive DW/BI development  
NOT  
data models
Take an Evolutionary Approach

NOT

Source: Spotify
Agile DW/BI Teams Work in Short Iterations

1 week or less: 13%
2 weeks: 65%
3 weeks: 15%
4 weeks: 7%
> 4 weeks: 1%

Disciplined Agile DW/BI teams deliver new data or reports every iteration

*Source: SA+A 2015 Agile State of the Art Survey
Pay Down Data Technical Debt

- Extract Transform Load (ETL) is often a quality band-aid
- Transformation is time consuming and risks defect injection
  - Often needed due to semantics differences
- Disciplined agilists strive to fix data problems at the source via database refactoring
  - This is one aspect of paying down technical debt
Agile DW/BI is Pragmatic

- Sometimes:
  - You need to accept data inconsistencies
  - There is no “one truth”
  - You can’t get all of the data (at first) that you need for a report
  - Stakeholders can’t agree

- Perfection is the enemy of good enough
Choose the Right Lifecycle

- When the concept of DW/BI is completely new to your organization
  - The Exploratory lifecycle should be considered so that you can identify a potential strategy for the DW/BI solution

- Initial release of a DW/BI solution
  - The Agile/Basic lifecycle is the typical choice
  - Straightforward approach to agile
  - Iterations force discipline onto a team that is likely new to agile

- Future releases of a DW/BI solution
  - The Lean or Continuous Delivery lifecycles are the typical choices
  - New requirements are often small and independent of one another
Create Light-weight, Valuable Artifacts
A Disciplined Agile Approach to DW/BI Development

**Inception**
- Come to agreement around high-level requirements, architecture, and plan
- Initiate the team
- Setup work environment and tools
- Secure funding for team

**Construction**
- Incrementally build the data warehouse
- Prove the architecture early
- Development of vertical, fully functional slices
- Detailed data source analysis
- Source-to-target data mapping (implementation)
- Database refactoring
- Physical data modeling
- Regression testing
- Continuous integration (CI)
- Continuous deployment (CD)
- Continuous documentation
- Detailed planning
- Coordination meetings
- (Iteration) demo
- Retrospectives

**Transition**
- Ensure the solution is ready to be deployed
- Deploy the solution
- End-of-lifecycle testing
- Last-minute fixes
- Finalize documentation
- Deploy database schema changes
- Migrate production data to new schema

**Primary Activities**
- Initial usage modeling (user stories and epics)
- Initial conceptual modeling (high level)
- Identification of potential data sources
- High-level data source analysis
- Initial architecture modeling
- Initial release planning
- Adopt common development and data conventions

**Secondary Activities**
- Logical data modeling
- Source-to-target data mapping
- Detailed data source analysis
- Logical data modeling
- Source-to-target data mapping (documentation)
- Meta-data documentation
- Finalize secondary documentation
The Agile Database Techniques Stack

- Vertical Slicing
- Clean Architecture and Design
- Agile Data Modeling
- Database Refactoring
- Database Regression Testing
- Continuous Database Integration
- Configuration Management
Vertical Slices of a Solution

• Every iteration a disciplined agile team produces a working solution
• Functionality is added in “vertical slices”

• Example slicing strategies for DW/BI:
  – One new data element from a single data source
  – One new data element from several sources
  – A change to an existing report
  – A new report
  – A new reporting view
  – A new data mart table

© Disciplined Agile Consortium
Considerations for Clean Data Architecture

• Consistency
  – ACID (Atomic Consistent Isolated Durable) transactions
  – Eventually consistent
  – Good enough
• Problem/Solution fit
  – SQL vs. Hierarchical vs. No-SQL vs. files
• Latency
  – How fast does access need to be?
• Security
  – Who should access which data?
• Scalability
  – Do you need to cluster?
  – Support multiple physical locations globally?
• Sourcing
  – Get the right data from the best sources
• Auditability
  – Ensure that you know where and when data came from
• Historical storage
  – Retain previous versions of appropriate data
Strategies for Clean Database Design

- **Normalization**
  - Are you storing data in one and one only place?

- **Design for the database type**
  - OLTP (transactional → normalized design)
  - OLAP (analytical → denormalized design)

- **Tables and columns should be cohesive**

- **Future proofing?**
  - Maintaining historical data values
  - Soft vs. hard deletes
  - Truly unique surrogate keys
  - Implement all relations as many-to-many

- **Set and follow naming conventions**
WE NEED DETAILED DATA MODELS FROM THE START OF THE PROJECT.

THAT'S TOO RISKY! WE NEED TO WORK IN A MORE AGILE MANNER.
Agile Data Modeling: Concept

- Data modeling is the act of exploring data-oriented structures
- Evolutionary data modeling is data modeling performed in an iterative and incremental manner
- Agile data modeling is evolutionary data modeling done in a collaborative manner

There is nothing special about data modeling!
Agile Data Modeling: In Practice

- Inception: Initial Modeling
  - High-level conceptual modeling

- Construction
  - JIT Modeling
    - Detailed physical data modeling (to generate DDL)
  - Test-Driven Development
    - Detailed specification
WHAT IF I TOLD YOU

THAT IT IS TRIVIAL TO EVOLVE A PRODUCTION DATABASE SCHEMA?
Why Evolving Databases is Thought to be Hard

Production databases are often highly coupled to other systems, services, data sources, …
Database Refactoring

A database refactoring is a simple change to a database schema that improves its design while retaining both its behavioral and informational semantics.

A database schema includes structural aspects such as table and view definitions; functional aspects such as stored procedures and triggers; and informational aspects such as the data itself.
The Process of Database Refactoring

1. Verify that a refactoring is needed
2. Choose the Right Refactoring
3. Deprecate the Original Schema (optional)
4. Run the tests
5. Write a Unit Test
6. Change your schema
7. Migrate Data (optional)
8. Update External Access Programs
9. Run the tests
10. Announce The Refactoring
11. Version Control Your Work

© Disciplined Agile Consortium
SO YOU'RE TELLING ME THAT DATA IS A CORPORATE ASSET

YET YOU DON'T HAVE REGRESSION TEST SUITES FOR YOUR DATA SOURCES?
Database Regression Testing

- Agilists develop regression unit and acceptance test suites for their applications, why not for databases too?
- If database testing is done at all these days, it is at the black-box level. This is a good start but often isn't enough
- You must be able to put the database into a known state, therefore you need test data (generation)
Test/Behavior Driven Database Development (TDD/B/DDD)

- An evolutionary approach to database design
- Driven by tests, not models
- A practical approach to ensuring database quality
- Completely different than traditional approaches to database design
- Really just one part of BDD, not a stand-alone activity

- May/June 2007 issue of IEEE Software
Continuous Integration and Databases

• Part of building the system is building the database (if it changed)

• Challenge: Tests SHOULD put the database back into a known state, but sometimes don’t
  • You will want to rebuild the (non-production) database from scratch every so often

• Challenge: Database accesses take time
  • Some test suites will test against DB mocks
  • You still need to test the actual database occasionally
I PITY THE FOOL

WHO DOESN'T HAVE CONTINUOUS INTEGRATION (CI) IN PLACE.
Configuration Management

- All work products should be stored in a versioned repository
- Maintains the integrity of the system and all supporting work products as it evolves
- Facilitates change in a controlled fashion
- All revisions kept and who made what changes to all work products
- Potential benefits include the ability to:
  - Manage versions of systems across environments
  - Rollback to previous versions
  - Modify work products in parallel and then merge
  - Find source of defects injected

© Disciplined Agile Consortium
Adoption Rates of Potential Data Quality Strategies

- ETL Transformation: 57%
- Logical Data Modeling: 54%
- Enterprise Data Modeling: 47%
- Data governance: 46%
- Master Data Management: 37%
- Data stewards: 31%
- Database regression testing: 31%
- Continuous database integration: 29%
- Database refactoring: 22%
- Test Driven DB Development: 16%

Source: 2015 Data Quality Survey
Parting Thoughts
THINK OUTSIDE THE BOX
Modern development is evolutionary in nature
You need to be flexible
BIG IDEA → DREAM → CHALLENGE 

PUSH YOURSELF

LEAVE YOUR COMFORT ZONE

TRY IT → CHANGE

GROW
Disciplined Agile DW/BI Workshops

• We offer a two versions of a DW/BI workshop
  – One-day: For people experienced with agile development who have been trying to apply it in the DW/BI space
  – Two-day: For teams new to Agile DW/BI

• Please visit DisciplinedAgileConsortium.org for information about Disciplined Agile workshops.
Thank You!

scott [at] scottambler.com
Twitter: @scottwambler

DisciplinedAgileConsortium.org
DisciplinedAgileDelivery.com
Would You Like This Presented to Your Organization?

Contact us at ScottAmbler.com
Scott Ambler + Associates is the thought leader behind the Disciplined Agile Delivery (DAD) framework and its application. We are a boutique IT management consulting firm that advises organizations to be more effective applying disciplined agile and lean processes within the context of your business.

Our website is ScottAmbler.com

We can help