Agile Enterprise Data Warehousing
Radical idea or practical concept?

Larissa T. Moss
Method Focus Inc.
methodfocus@earthlink.net

TDWI • South Florida Chapter • March 11, 2011
Outline

- Why traditional methodologies don’t work on DW/BI projects
- Agile BI versus Agile DW
- What works and what doesn’t on DW projects
- Extreme Scoping™ Agile Approach to DW/BI
DW/BI: From Chaos to Architecture
Waterfall Methodologies

Two problems:
1. No data integration effort
2. Takes too long to deliver
Spiral Data Integration Methodologies

Solve problem 1

Assessment & Strategy → Data Requirements

Business Opportunity

Post-Impl. Review

Integration Analysis

Implementation

Testing

Development

Evolving Data Warehouse

80% Data Management

20% Data Delivery

Iterative Development = One application at a time
Agile Development Methodologies

Solve problem 2

One Application = Multiple Releases

© Copyright 2011, Larissa T. Moss, Method Focus, Inc.
Can Agile be used for BI?

- Depends on what you call “BI”
- Depends on what you call “Agile”

Solve problem 2

- Scrum XP
- FOCUS on: Coding apps

• Independent BI applications
• Directly sourced from operational databases
• Developed by the same or by different BI groups

• Independent BI applications
• Sourced through a central staging area database containing copies of operational databases
• Developed by the same or by different BI groups

© Copyright 2011, Larissa T. Moss, Method Focus, Inc.
Can Agile be used for BI?

- Depends on what you call “BI”
- Depends on what you call “Agile”

Solve problem 1

BI with separate DW

- Dependent BI applications
- Sourced from an integrated and standardized DW
- Developed by the same or by different BI groups
- But, building/enlarging the DW is a different project and a different team
Can Agile be used for BI?

- Depends on what you call “BI”
- Depends on what you call “Agile”

Solve problems 1&2

- Scrum
- XP

- Dependent BI applications
- Sourced from an integrated and standardized DW or from dependent data marts
- Developed by the same group
- Building or enlarging the DW is the same project and the same team

© Copyright 2011, Larissa T. Moss, Method Focus, Inc.
Can Agile be used for BI?

- Depends on what you call “BI”
- Depends on what you call “Agile”

- Dependent BI applications
- Sourced from an integrated and standardized DW or from dependent data marts
- Developed by the same group
- Building or enlarging the DW is the same project and the same team

**Solve problems 1&2**

**Extreme Scoping™**

**FOCUS on:**
- Enterprise data standardization and integration
- Not: Coding apps

**BI includes DW**

© Copyright 2011, Larissa T. Moss, Method Focus, Inc.
No “Silo” and No “Big-Bang” Development

Solve problems 1 & 2

Merge Spiral with Agile!

= Extreme Scoping™
Agile principles that work for DW projects

- Business vision instead of final requirements
- Speculation instead of estimating
- Exploration (prototyping) instead of development
- Self-organizing project team (no interference)
- Daily stand-up meetings (course corrections)
- Get physical as quickly as possible (deliver partial functionality)
- Time-boxed increments
- (Data) Quality before quantity
- Refactoring (refinement = system evolution)

4 E’s:
- Experimental
- Experiential
- Educational
- Evolutionary
Agile principles that *don’t work* for DW projects

- Freezing the scope of an increment
- Scrum Master (project manager)
- Product owner (single user)
- Product backlog managed by product owner
- Effort estimate based on code complexity
- Cadence = development rhythm (every 10 or 29 days)
- Burn-down charts
Step 1: Speculation and Scouting

(Trademark of Larissa T. Moss, Method Focus Inc.)

1. Speculate on the total effort for the new DW/BI request

Development Steps
1. Business Case Assessment .................................. Enterprise-view
2.B Non-Technical Infrastructure Evaluation ... Enterprise-view
3. Project Planning .............................................. Enterprise-view
4. Requirements Definition ................................... Enterprise-view
5. Data Analysis ................................................. Enterprise-view
6. Application Prototyping ..................................... Enterprise-view
7. Meta Data Repository Analysis ......................... Enterprise-view
8. Database Design ............................................. Enterprise-view
9. ETL Design ..................................................... Enterprise-view
10. Meta Data Repository Design ............................ Enterprise-view
11. ETL Development ............................................ Enterprise-view
12. Application Development ................................ Enterprise-view
13. Data Mining .................................................. Enterprise-view
14. Meta Data Repository Development ................. Enterprise-view
15. Implementation .............................................. Project-specific
16. Release Evaluation ......................................... Enterprise-view

From Business Intelligence Roadmap, Moss Larissa T. and Shaku Atre)

“Speculation”

Selected things to consider
• activities
• dependencies
• deliverables
• roles

Unknown conditions (Questions)
• data
• architecture
• inter-dependencies
• resources

“Scouting”

© Copyright 2011, Larissa T. Moss, Method Focus, Inc.
Speculation and Scouting Example

Development Steps

1. Business Case Assessment .......................  
2. Technical Infrastructure Evaluation ...........
2.1 Non-Technical Infrastructure Evaluation ... 5 days
3. Project Planning .....................................
4. Requirements Definition ........................... 15 days
5. Data Analysis .....................................
6. Application Prototyping ............................. 10 day
7. Meta Data Repository Analysis .................. 1 day
8. Database Design ..................................... 22 days
9. ETL Design ........................................... 15 days
10. Meta Data Repository Design .................... 6 days
11. ETL Development ................................... 45 days?
12. Application Development .......................... 10 days
13. Data Mining ........................................
14. Meta Data Repository Development ............ 10 days
15. Implementation ..................................... 8 days
16. Release Evaluation ................................. 1 day

“Speculation”

Selected things to consider
- activities
- dependencies
- deliverables
- roles

- Identify data sources
- Identify data owners
- Profile data sources
- Create dirty data report
- Write cleansing specs

Unknown conditions (Questions)
- data
- architecture
- inter-dependencies
- resources

“Scouting”

~ 148 days + ?

~ 10 days
Speculation and Scouting Example

Development Steps

1. Business Case Assessment
2. A Technical Infrastructure Evaluation
2.B Non-Technical Infrastructure Evaluation ...
3. Project
4. Requirements ~ 60 elapsed weeks
5. Data Analysis
6. Application Prototyping
7. Meta Data Repository Analysis
8. Database Design
9. ETL Design
10. Meta Data Repository Design
11. ETL Development
12. Application Development
13. Data Mining
14. Meta Data Repository Development
15. Implementation
16. Release Evaluation

~ 56 elapsed weeks
~ 10 days

“Speculation”

Selected things to consider
- activities
- dependencies
- deliverables
- roles

~ 148 days + 17

- Identify data sources 1 day
- Identify data owners 1 day
- Profile data sources 8 days
- Create dirty data report 2 days
- Write cleansing specs 5 days

Unknown conditions (Questions)
- data
- architecture
- inter-dependencies
- resources

“Scouting”

~ 10 days
Step 2: Extreme Scoping™ Applied

1. Speculate on the total effort for the new DW/BI request
2. Break DW/BI application request into software releases

BDTP™ Balance!

<table>
<thead>
<tr>
<th>I</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Value</td>
<td>Data Efforts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Considerations</td>
<td>Project &amp; Program Interdependencies</td>
</tr>
</tbody>
</table>
BDTP Balance™

1. Business representative chooses the function (feature) with the highest *business value*

2. EIM staff estimates the *data effort* to determine scope and deadline (time-box)

3. Technical lead(s) determine *technical feasibility* and technology readiness

4. Project manager examines *intra-project inter-dependencies*

5. Program manager determines *inter-project inter-dependencies*
How Many Software Releases?

180 effort days = 60 elapsed weeks (~14 months)

Data Scope?
R1: ~ 2 months
R2: ~ 4 months
R3: ~ 4 months
R4: ~ 2 months
R5: ~ 2 months

180 effort days = 60 elapsed weeks (~14 months)
Step 3: Planning the First Release

1. Speculate on the total effort for the new DW/BI request
2. Break DW/BI application request into software releases
3. Create work breakdown structure for first software release
## Select Appropriate Tasks or Subtasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>592 Write query script specifications</td>
<td>587</td>
</tr>
<tr>
<td>593 Write access interface programming specifications</td>
<td>588</td>
</tr>
<tr>
<td>594 Write online help function programming specifications</td>
<td>589</td>
</tr>
<tr>
<td><strong>Step 11: Extract/Transform/Load Development</strong></td>
<td></td>
</tr>
<tr>
<td>595 <strong>Build and unit test the ETL process</strong></td>
<td>542, 549</td>
</tr>
<tr>
<td>596 Code the ETL programs</td>
<td></td>
</tr>
<tr>
<td>597 If using an ETL tool, write instructions for ETL tool modules</td>
<td></td>
</tr>
<tr>
<td>598 Capture the ETL technical metadata for the metadata repository</td>
<td></td>
</tr>
<tr>
<td>599 Write code to produce reconciliation totals, quality metrics, and load statistics</td>
<td>597, 600</td>
</tr>
<tr>
<td>600 Unit test each individual program module</td>
<td>598, 600</td>
</tr>
<tr>
<td>601 If using an ETL tool, unit test each ETL tool module</td>
<td></td>
</tr>
<tr>
<td>602 Write the scripts to execute the ETL programs and the sort, merge, and load utilities</td>
<td>601, 602</td>
</tr>
<tr>
<td><strong>604 Integration or regression test the ETL process</strong></td>
<td>596</td>
</tr>
<tr>
<td>605 Create a test plan with test cases for the ETL process</td>
<td></td>
</tr>
<tr>
<td>606 Create test data for the ETL programs</td>
<td></td>
</tr>
<tr>
<td>607 Integration or regression test the entire ETL process</td>
<td>606, 606</td>
</tr>
<tr>
<td>608 Log the actual test results and document any test issues</td>
<td></td>
</tr>
</tbody>
</table>

*from Business Intelligence Roadmap (Moss & Atre)*
Step 4: Weekly milestones

1. Speculate on the total effort for the new DW/BI request
2. Break DW/BI application request into software releases
3. Create work breakdown structure for first software release
4. Create milestones from DDD to YAH for first release

(Trademark of Larissa T. Moss, Method Focus Inc.)
### Describing the milestones

**Week 1:**
- Potential source files identified and profiled
  - Data cleansing specifications identified and reviewed with users
  - ETL architectural design changes identified
  - 8 report designs prototyped and demonstrated to users

**Week 2:**
- New data modeled and data disputes among users documented
  - New table created and 3 existing databases modified
  - Undisputed data mapped from source to target
  - One new extract program coded and 5 ETL programs modified

**Week 3:**
- All data disputes among users resolved
  - Data cleansing specifications modified
  - All ETL programs unit tested
  - 5 new reports coded based on approved designs using new table
  - 3 old reports from existing databases modified

**Week ddd-n:**
- ...

**Week ddd-2:**
- UAT completed and code frozen
  - Operations signed off
  - Job scheduler created and tested
  - Production environment created

**Week ddd-1:**
- All programs migrated into production environment
  - Load files ready to run in production
  - Job scheduler modified, tested, and signed off
Step 5: Parallel Development Tracks

1. Speculate on the total effort for the new DW/BI request
2. Break DW/BI application request into software releases
3. Create work breakdown structure for first software release
4. Create milestones from DDD to YAH for first release
5. Organize and assign parallel development tracks
Parallel Development Activities

Justification

1 Bus. Case Assess.

Go/No-Go decision

Project kick-off

Planning (Analysis)

2 Enterprise Infrastr.

3 Project Planning

4 Requirement Definition

Analysis

5 Data Analysis

6 Prototyping

9 ETL Design

11 ETL Develop.

Design

8 DB Design

12 Application Develop.

Construction

13 Data Mining

Deployment

15 Implement.

16 Release Eval.

Development Track Teams

7 MDR Analysis

10 MDR Design

14 MDR Develop.

Meta Data Rep.
Step 6: Informal Internal Micro Plan

1. Speculate on the total effort for the new DW/BI request
2. Break DW/BI application request into software releases
3. Create work breakdown structure for first software release
4. Create milestones from DDD to YAH for first release
5. Organize and assign parallel development tracks
6. Create detailed work assignments for weekly milestones

A unit of work that may be measured in days or hours produces managed Task Deliverable

© Copyright 2011, Larissa T. Moss, Method Focus, Inc.
Example: Task Board
Step 7: Formal Management Macro Plan

1. Speculate on the total effort for the new DW/BI request
2. Break DW/BI application request into software releases
3. Create work breakdown structure for first software release
4. Create milestones from DDD to YAH for first release
5. Organize and assign parallel development tracks
6. Create detailed work assignments for weekly milestones
7. Create milestone chart for progress reporting

<table>
<thead>
<tr>
<th>Milestone Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wk 1:</strong></td>
</tr>
<tr>
<td>- Data cleansing specifications identified and reviewed with users</td>
</tr>
<tr>
<td>- ETL architectural design changes identified</td>
</tr>
<tr>
<td>- Report designs proposed and demonstrated to users</td>
</tr>
<tr>
<td><strong>Wk 2:</strong></td>
</tr>
<tr>
<td>- New table created and 3 existing databases modified</td>
</tr>
<tr>
<td>- Undisputed data mapped from source to target</td>
</tr>
<tr>
<td>- One new extract program coded and 5 ETL programs modified</td>
</tr>
<tr>
<td><strong>Wk 3:</strong></td>
</tr>
<tr>
<td>- Data cleansing specifications modified</td>
</tr>
<tr>
<td>- All ETL programs unit tested</td>
</tr>
<tr>
<td>- 5 new reports coded based on approved designs using new table</td>
</tr>
<tr>
<td>- 3 old reports from existing databases modified</td>
</tr>
<tr>
<td><strong>Wk ddd-n:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Milestone Description</th>
<th>Oct. 8</th>
<th>Oct. 15</th>
<th>Oct. 22</th>
<th>Nov. 5</th>
<th>Nov. 12</th>
<th>Nov. 19</th>
<th>Nov. 26</th>
<th>Dec. 3</th>
<th>Dec. 10</th>
<th>Dec. 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest level of project partitioning</td>
<td>milestone</td>
<td>describes</td>
<td>State of project reported</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Copyright 2011, Larissa T. Moss, Method Focus, Inc.
### Example: Milestone Chart

<table>
<thead>
<tr>
<th>Milestone Description</th>
<th>Oct. 8</th>
<th>Oct. 15</th>
<th>Oct. 22</th>
<th>Oct. 29</th>
<th>Nov. 5</th>
<th>Nov. 12</th>
<th>Nov. 19</th>
<th>Nov. 26</th>
<th>Dec. 3</th>
<th>Dec. 10</th>
<th>Dec. 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wk 1: - Potential source files identified and profiled</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>- Data cleansing specifications identified and reviewed with users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ETL architectural design changes identified</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 8 report designs prototyped and demonstrated to users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wk 2: - New data modeled and data disputes among users documented</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>- New table created and 3 existing databases modified</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Undisputed data mapped from source to target</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- One new extract program coded and 5 ETL programs modified</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wk 3: - All data disputes among users resolved</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>- Data cleansing specifications modified</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- All ETL programs unit tested</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 5 new reports coded based on approved designs using new table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 3 old reports from existing databases modified</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wk ddd-n: ...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wk ddd-2: - UAT completed and code frozen</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>- Operations signed off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Job scheduler created and tested</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Production environment created</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wk ddd-1: - All programs migrated into production environment</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
</tr>
<tr>
<td>- Load files ready to run in production</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Job scheduler modified, tested, and signed off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- ▲ Completed on time
- ▲ Completed with significant delay
- ▲ Not completed yet
- ▲ Major modification to milestone (see detailed explanation on next page)
Planning the Next Software Release

1. Review the total effort for the new DW/BI request
2. Review and adjust DW/BI application software releases
3. Create work breakdown structure for next software release
4. Create milestones from DDD to YAH for next release
5. Organize and assign parallel development tracks
6. Create detailed work assignments for weekly milestones
7. Create milestone chart for progress reporting
Agile Enterprise Data Warehousing
Radical idea or practical concept?

- Depends on what you call “BI” (see BI Maturity Model)
- Depends on what you call “Agile”

Source: BI Maturity Model, Wayne Eckerson, TDWI
Thank You