Agile Methods and Data Warehousing

Kent Graziano
Enterprise Data Integration Team
Department of Technology Services
Denver Public Schools
Denver, Colorado USA

Agenda

- Our Bios
- Assumptions and Objectives
- Agile Manifesto
- Agile Principles
- Agile Concepts and Methods
- Two Week Iterations
- Getting to Agile/RAD
- Data Vault
- Conclusion

Our Bios

- Kent Graziano
  - Manager, Enterprise Data Integration
  - Oracle Designer and Data Warehouse Specialist
  - Co-Author of
    - The Data Model Resource Book (1st Edition)
    - Oracle Designer: A Template for Developing an Enterprise Standards Document
  - Past-President of ODTUG and RMOUG

Assumptions

- A beginner presentation
  - You have little experience with Agile Methods
- Not a beginner presentation
  - You already understand and have experienced the pain of a data warehouse project

Objectives

- Understand what is meant by "agile"
- Try to apply some agile ideas to data warehouse and business intelligence efforts
- Answer the question – can we deliver results faster?

Manifesto for Agile Software Development

http://agilemanifesto.org

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan
That is, while there is value in the items on
the right, we value the items on the left more

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Principle #1

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - Who is the customer?
  - What is "valuable software" in data warehousing?
    - BI reports
    - Dashboard interface
    - Working ETL code?

Principle #2

- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - Not much choice these days!
  - Must be flexible and adaptable in thinking and design
  - Use code generators
  - Start with normalized models

Principle #3

- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - Need good scope control!
  - One subject area at a time (Feature or Feature Set)
    - What is a subject area? What is a feature set?
  - Think Data Vault

Principle #4

- Business people and developers must work together daily throughout the project.
  - Great idea! (duh)
  - DW MUST have the business involved
    - One of the Top 10 reasons for failure
  - This applies for BI reports
    - Daily interaction would be great!
  - But – politics and priorities may interfere!

Principle #5

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - Wouldn't this be nice…
  - Need people who WANT to be on the project
  - Get training if needed
  - Keep units of work small to create an atmosphere of success
  - Don't try a Big Bang EDW

Principle #6

- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Developers hate documentation
  - Daily team huddles (< 30 min. daily)
  - While efficient, still need some documentation (or meta-data) for later

Principle #7

- Working software is the primary measure of progress.
  - Again – duh!
  - Applied to DW:
    - What is "working software?"
      - BI reports
      - Tables definitions and working ETL code
    - Think more broadly – it is not just a data entry screen

Principle #8

- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - DW projects last a long time – don't burn the team out with unreasonable deadlines
    - CIF is an architecture not a project!
  - See P#5 – Motivated individuals
  - Good planning and scope control
    - No all nighters!
  - Smallest valuable unit of work possible
    - Keep it moving like a production line
  - Pick (or develop) a standard, repeatable methodology
    - Study the Agile methods and adopt what works for your team
  - XP and Pair Programming aka Shoulder Surfing

Principle #9

- Continuous attention to technical excellence and good design enhances agility.
  - Bad design + bad architecture = trouble
    - Symptom: can't build a requested data mart
  - Frequent design reviews a must
    - Improves team skills – provides cross training
    - Over time – better designs, shorter review cycles
      - Faster delivery

Principle #10

- Simplicity--the art of maximizing the amount of work not done--is essential.
  - KISS – Keep it Simple Stupid
  - Write less code by hand
    - Use code generators! (No syntax errors – ever)
    - Modifications are easier – just regenerate the code
  - Use your meta data effectively
    - Diagrams can be used to review with business
    - Designer ROB – Repository Object Browser
      - Online data dictionary

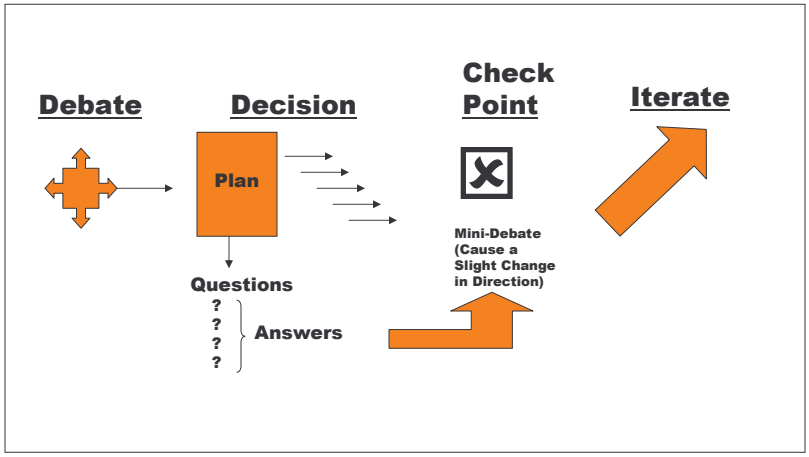
Principle #11

- The best architectures, requirements, and designs emerge from self-organizing teams.
  - Team of smart, motivated people = success
    - At DPS – We succeed (or fail) as a TEAM
  - Don't micro manage or pigeon-hole staff
    - Encourage team work and team thinking
    - Staff will gravitate to roles based on skills, interest, and personality
      - Then they have more buy-in to the process
    - Eliminates delays and bottlenecks by having shared responsibilities (no single point of failure)

Principle #12

- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
  - At DPS – The Decision Model
    - Debate Mode
    - Check Points
  - TQM anyone?
  - Related to self-organizing teams
    - Make finding the solution to a problem the team's problem
      - More buy-in to the solution

DPS Decision Model



Agile Concepts and Methods

- Feature Driven Development
- Team Huddles
- Extreme Programming
- Pair Programming

Feature Driven Development

- One of many Agile methods
  - Seems most applicable to DW
- Delivery of Features and Feature Sets
  - Feature = One Data Mart Report
  - Feature Set = DW Subject Area?
  - Feature Set = Star Schema Data Mart
- FDD Subject Area = Project = Group of Feature Sets
- Goal: Deliver a Feature Set quickly

Feature Driven Development

- How big should a subject area be for a DW project?
  - Customer
  - Product
  - Sales
- These are too big to be agile – need a new way of thinking

Feature Driven Development

- Sub-subject Areas
  - One main entity
  - Attributes
  - Related reference entities
- Must facilitate delivery of a specific query or report required by the business
  - Must deliver something of value (see P#1)
- Goal: Deliver model in smaller chunks
  - Get models to ETL programmers more frequently
  - Shorter iterations, faster deliverables = perceived progress
  - Traditional dw subject areas = lack of visible progress

Team Huddles

- Daily Standup Meeting - Morning Roll Call (FDD)
- Short meeting (< 30 minutes)
  - Every morning, mandatory attendance
  - Review assignments, accomplishments, backlogs
- Immediate feedback and assistance
  - Keeps team motivated and on track (P #5)
  - Identifies constraints and bottlenecks early in the process
  - Eliminates backlogs more quickly via re-assignments
- Improves team work
- Supports self-organizing teams (P #11)

Extreme Programming (XP)

- Programmer works directly with the end user
- In DW:
  - Best with developing BI reports
  - DW or data mart must already be populated
  - Reports developed using BI tool
    - With the user in the room
    - With constant user reviews and input using a web reporting tool
  - Also applies to developing a dashboard or portal interface

Pair Programming

- Part of XP
- Programmers work side-by-side
  - One terminal
    - One codes, the other reviews
  - Two terminals, one cube
    - One programming, one documenting
- In DW:
  - Writing ETL Code
  - Pair data modeling
  - Pair code promotion

Two week iterations?

- Goal is really a few weeks to a few months (see P#3)
- What is the deliverable?
  - A fact table for a star schema
  - A dimension table
  - A complete star (fact and all dimensions)
  - One piece of ETL code that populates a fact table
  - A function needed by the ETL code
  - A new report or query
- Who is the customer?
  - BI programmer?
  - Knowledge worker?
  - ETL programmer?

Getting to Agile/RAD

- "better than average expertise"
  - Expert consulting and mentoring
  - Do the work (OTJ)
- At DPS – took two years before we could try being more "agile"
  - Needed experience in DW, Designer, OWB, and the "process" of building, deploying, and maintaining an Oracle DW

Data Vault

- New modeling technique for enterprise data warehouse design
  - Best of breed technique blending 3NF and Star Schema
  - Data Vault white papers at www.danlinstedt.com
- Allows modeling EDW in small chunks
  - Develop model, build tables, build ETL, populate, repeat (often)
  - Key: prioritize the data requirements

References

- Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results by David J. Anderson
- The Goal by Eliyahu M. Goldratt
  - Recommended by Rachael Carmichael
- CASE Method Fast-track: A RAD Approach by Richard Barker & Dai Clegg
- Data Vault white papers at www.danlinstedt.com

Conclusion

- Agile concepts can be applied to data warehouse and BI projects
  - Not a purist definition!
  - Try to apply the principles – be creative
- Suggested approaches
  - Use team huddles
  - Use universal models as template
  - Use pair programming to increase quality and cross training
  - Use code generators like Designer & OWB
  - Use the Data Vault modeling approach
  - Read about Agile Methods (XP & FDD)
  - Read Oracle CASE Method fast-track
  - Be flexible and give it a try

Questions?

Contact Information

- Kent Graziano
  - Kent_Graziano@dpsk12.org