



Agile Data Warehousing Project Management With SCRUM HDT823 Three Days

Prerequisites

Students should have experience designing databases and data warehouses. Knowledge of Agile Scrum and Extreme Programming techniques is helpful.

Who Should Attend

This course is targeted at business intelligence developers and users, data warehouse developers, star schema and cube developers, data modelers, database analysts, and any other personnel who need to manage a business intelligence project.

Course Description

This course provides students with the skills necessary to manage the design and implementation of a data warehouse using Agile Scrum and Extreme Programming techniques. The course features the six-stage plan, developed by Ceregenics, Inc, for launching Agile data warehousing projects. Agile Data Warehousing not only offers data professionals financial savings but also shortens delivery times and improves application quality. Agile Data Warehousing is based on the following book: [Agile Data Warehousing: Delivering World-Class Business Intelligence Systems Using Scrum and XP](#) by Ralph Hughes published by IUniverse on August 5, 2008, ISBN: 0595471676.

Course Topics

- Agile Data Warehousing (ADW) Motivation and Overview
- Generic Scrum In a Nutshell
- User Stories For Agile Warehousing
- Avoiding Over-commitment With Agile Estimation
- Adapting Scrum For Data Warehousing
- Starting and Scaling Agile Warehousing Teams
- ADW As a Demonstrably Mature Method
- Managing Adversity



Agile Data Warehousing Project Management With SCRUM HDT823

I. Agile Data Warehousing (ADW) Motivation and Overview

A. Lamentable Patterns In Waterfall BI Projects

1. Software Development Yields Notoriously Poor ROI
2. Customers Are Increasingly Impatient to Get BI Benefits
3. BI Teams Cannot Avoid Tackling the Unknown
4. Infrastructure Problems Cause Problems Late in the Game
5. Business Rules Can Also Appear Late in the Plan
6. Projects Take Fatal Risks in Order to Avoid a Little Rework
7. Specification and Review Cycles Consume Excessive Effort
8. To-Do Documents Have Too Short a “Shelf Life”
9. Project Management Consumes Large Portions of the Budget
10. Complex Methods and Elaborate Plans Rarely Perform Well
11. BI Teams Need More Time to Acclimate to a Project
12. Customers Need More Time to Learn About BI

B. Goals For a Better Method

C. A Quick Outline of Agile Data Warehousing

1. Stage 0—Colocated, Self-Organized Teams
2. Stage 1—User Epic Decomposition and Estimation Skills
3. Stage 2—Release Planning
4. Stage 3—Reference Models and Test-Led Development
5. Stage 4—Pipelined Delivery Squads
6. Stage 5—Continuous Integration Testing

D. Tracking the Maturity of ADW Teams

E. Not a Revolution, Just an Impressive Evolution

F. Subsequent Volumes for Engineering and Automation

G. Caveat Emptor

II. Generic SCRUM In a Nutshell

A. Iterations and Stories

1. The Release Cycle
2. The Development Cycle
3. The Daily SCRUM

B. Details Regarding the Generic Sprint

1. Story Conference
2. Task Planning
3. Development Phase
4. Self-Organization
5. Daily SCRUMS
6. 2-to-1 Design and Test Led Development
7. Architectural Compliance and “Tech Debt”



Agile Data Warehousing Project Management With SCRUM HDT823

- 8. User Demo
- 9. Sprint Retrospective
- C. Generic Tracking Tools and Project Visibility
 - 1. The Sprint Task Board
 - 2. The Burndown Chart
 - 3. Seeing Scope Creep
 - 4. Seeing Progress for an Entire Release
- D. The SCRUM Master
- E. Non-Standard Sprints
 - 1. Sprint 0
 - 2. Spikes
- F. Where Did SCRUM Come From?
 - 1. Distant History
 - 2. SCRUM Emerges
- G. Waterfall Methods—A Mistake From the Start
- H. Why Does SCRUM Work So Well?
 - 1. Avoids Over-Investment in Fragile Artifacts
 - 2. Connects Developers with Real User Requirements
 - 3. Eliminates Many Ineffective Meetings
 - 4. Increases Effective Planning
 - 5. Increases the Accuracy of Estimates
 - 6. Transparent to the Customer and Management
 - 7. Eliminates 1,000 Line Project Plans ...
 - 8. Making the Project Less Expensive to Manage
 - 9. Elevates Quality to a Daily Issue for Developers
 - 10. Incorporates Continuous Improvement
 - 11. Adapts Tools from Other Methods as Appropriate
 - 12. Leaves Most EIS “Best Practices” In Place
 - 13. Allows Customers to Change Their Minds
 - 14. Quick to Teach
 - 15. Surfaces the Problems Early On ...
 - 16. So There are Fewer Surprises Upon Implementation
 - 17. Results in Fewer Support Costs When Deployed
 - 18. Makes Cost Overruns Far More Manageable
- I. SCRUM Addresses Stakeholders’ Fears
 - 1. Customer Fears
 - 2. Developer Fears
 - 3. IT Management Fears
 - 4. Why Live with All This Fear?



Agile Data Warehousing Project Management With SCRUM HDT823

- III. User Stories for Agile Warehousing**
 - A. Traditional Requirements and Its Discontents**
 - B. The Generic SCRUM User Story**
 - 1. Generic Stories, Epics, and Themes
 - 2. Still a Seemingly Enormous Undertaking
 - C. Verbalizing User Stories**
 - 1. Diagnosis Phase
 - 2. Research Phase
 - 3. Decomposition Phase
 - 4. Prioritize Phase
 - 5. Build Phase
 - 6. Review Phase
 - 7. Sprint/Verbalization Integration
 - D. Decomposition for Back-End Epics**
 - 1. Target Layer Decomposition
 - i. Staging Layer
 - ii. Integration Layer
 - iii. Reference Tables
 - iv. Fundamental Tables
 - v. Linking Tables
 - vi. Linking History Tables
 - 2. Presentation Layer
 - i. Non-Historical Dimension Tables
 - ii. Historical Dimension Tables
 - iii. Fact Tables
 - 3. End User Access Layer
 - E. Refresh Type Decomposition**
 - 1. Direct Link
 - 2. Snapshots
 - i. Manually Invoked Refresh Subtypes
 - ii. Schedule-Driven Refresh Subtypes
 - iii. Error Recycling
 - 3. Incremental Loads
 - i. Incremental Load Refresh Subtypes
 - F. Refresh Frequency Decomposition**
 - G. Transformation Type Decomposition**
 - 1. Direct Data Transfer
 - 2. Aggregations
 - 3. Applied Business Rules
 - H. Decomposition for Front-End Epics**
 - 1. User-Friendliness Decomposition
 - i. Single-Table Access



Agile Data Warehousing Project Management With SCRUM HDT823

- ii. Modeled Access
- iii. Defined Navigation
- iv. Pick List Supported
- v. Dashboards
- 2. Automation Levels Decomposition
 - i. On-Demand Operating On User Workstations
 - ii. On-Demand Posted to Server
 - iii. Scheduled Refresh On Server
- I. Value Chains Product Owners Can Understand
 - 1. A Sample Back-End Value Chain
 - 2. A Sample Front-End Value Chain
- J. A Utility for Managing User Stories
 - 1. A Single Table Data Model
 - 2. Columns for the User Story Table

IV. Avoiding Over-Commitment with Agile Estimation

- A.** The Damage Done by Bad Estimates
- B.** Two Estimating Approaches
- C.** Why Waterfall Teams Underestimate
 - 1. A Single-Pass Effort
 - 2. Few Immediate Consequences and Feedback
 - 3. Over-Optimism Rewarded
 - 4. No Guard Rails Against Errors
- D.** The ADW Estimation Technique
 - 1. The Estimation Cycle
 - 2. Forecast Step
 - 3. Confirmation Step
 - 4. Watch Step
 - 5. Learn Step
- E.** Defer Release Planning for New Teams
- F.** Story Points Versus Ideal Time
 - 1. Story Points Defined
 - 2. Ideal Time Defined
 - 3. The Big Advantage to Story Points
 - 4. Quick Story Points Via “Estimation Poker”
 - 5. Labor-Hour Estimates Still Valuable
 - 6. Ensuring Consistency of Estimates
 - 7. Re-estimating Stories After Release Planning
- G.** Tracking Progress and Measuring Velocity
 - 1. Burn Down Charts
 - 2. Typical Patterns and Their Causes
 - i. An Early Hill to Climb



Agile Data Warehousing Project Management With SCRUM HDT823

- ii. Persistent Under-Delivery
 - iii. Fine-Tuning Daily Stand-Up Questions
 - H. Deriving Velocity in Original Labor Estimates
 - 1. Micro-Adjusting Velocity for the Next Sprint
 - 2. Setting a Working Velocity for a New Team
 - I. Dealing with Mid-Iteration Scope Creep
 - J. Velocity as the True Metric
 - 1. Maintaining a Meaningful Velocity
- V. **Adapting SCRUM for Data Warehousing**
 - A. Tiered Data Model
 - 1. Easier Scoping Decisions for Units and Measures
 - i. Seeing the ETL Units Needed
 - ii. Seeing the ETL Models Needed
 - 2. Easier Progress Tracking
 - 3. Balancing Design Clarity and ETL Performance
 - B. Agile Quality Assurance Overview
 - 1. Quality From Patterns and Reference Models
 - 2. Amassing Patterns for the Reference Model
 - 3. Getting the Most Out of a Reference Model
 - i. Better Level-of-Effort Estimates
 - ii. A Detailed “Definition of Done” for the Developers
 - iii. Faster As-Built Documentation
 - iv. Better Crafted Code Promotion Procedures
 - v. A Superior Tool for Testing Upgrades to BI Tools
 - vi. A Better Training Program for New Developers
 - 4. Quality Through Test-Led Development
 - i. Working with Test-Led Development
 - C. Quality Through Automated Testing
 - 1. ADW Testing Requirements
 - i. Regarding Quality
 - ii. Regarding Complexity
 - iii. Automated Test and Complex Validation
 - iv. Testing Business Rules with Row Triplets
 - 2. The Test Data Repository
 - 3. Automated Testing Facility Seen as a Whole
 - 4. Applying ACIT to Front-End Applications
 - 5. Summary of ACIT Benefits
 - D. Pipelined Delivery Squads
 - 1. Generic SCRUM Only a Partial Solution
 - 2. The Structure of an ADW Pipeline
 - i. Informally-Defined Work Squads



Agile Data Warehousing Project Management With SCRUM HDT823

- ii. Should ETL and OLAP Be Different Squads?
- iii. Maintaining a Clear Notion of Team Velocity



Agile Data Warehousing Project Management With SCRUM HDT823

- VI. Starting and Scaling Agile Warehousing Teams**
 - A. Primary Leadership Roles**
 - 1. Product Owner
 - 2. Solutions Architect
 - 3. Data Architect
 - B. Developer Roles**
 - 1. Senior OLAP Developer
 - 2. Integration and Testing Engineer
 - C. ETL and OLAP Developers**
 - D. Methodological Support Roles**
 - 1. SCRUM Master
 - 2. Project Communications Assistant
 - E. Test-Led Development as a Shared Responsibility**
 - F. Nurturing a Top-Notch ADW Team**
 - 1. Six Stages Overview
 - 2. Stage 0: Generic SCRUM
 - 3. Stage 1: User Epic Decomposition
 - 4. Stage 2: Formulating the Release Plan
 - 5. Stage 3: Reference Models and Test-Led Development
 - 6. Stage 4: Pipelined Delivery Squads
 - 7. Stage 5: Implement the Automated Testing Facility
 - G. Scaling Up with a SCRUM of SCRUMs**
 - 1. Recommendations from the Agile Community
 - 2. Early Iterations Will Frustrate Program Sponsors
 - 3. Decoupling BI Efforts
 - 4. Re-Couple Teams Through ACIT for Quality Assurance
 - 5. SCRUM Too Light for Scaling
 - 6. “Critical Chain” for Program Management
 - 7. Reveals Unrealistic Deadlines Through Resource Leveling
 - 8. Brings Deadlines in by Pooling Task Padding
 - 9. Protects the Critical Path with Three Types of Buffers
 - 10. Accelerates Projects Through Changes in Work Culture
 - 11. Managing Projects Focuses Upon Buffer Recovery
 - 12. Adapting Critical Chain for Scaling SCRUM
 - H. Overcoming Geographical Barriers**
 - 1. Choose the Geographical Division Carefully
 - 2. Creatively Define Roles to Mitigate Separation
 - 3. Budget for Top-Of-Sprint Travel
 - 4. Insist on Collocation for Two or More Sprints
 - 5. Agile Collaboration Software



Agile Data Warehousing Project Management With SCRUM HDT823

- VII. ADW as a Demonstrably Mature Method**
 - A.** Why the Capability Maturity Model (CMM)?
 - B.** CMM Overview
 - C.** Calmly Countering the “Immaturity” Label
 - D.** Point 1—CMM May Be Overkill
 - 1. Deciding Upon Plan-Driven or Adaptive Methods
 - E.** Point 2—CMM Not as Rigid as Many Suppose
 - 1. Project Planning
 - 2. Requirement
 - 3. Design
 - 4. Validation and Verification
 - 5. Product Integration
 - F.** Point 3—Keep Approaches and Methods Distinct
 - G.** Point 4—Multiple Agile Methods CMM Compliant
 - 1. Microsoft’s Solutions Framework
 - 2. AgileTek’s Agile+
 - 3. BoldTech’s BoldDelivery
 - H.** Point 5—ADW Contains What CMM Requires
 - 1. Project Planning
 - 2. Project Management and Control
 - 3. Supplier Agreement Management
 - 4. Integrated Project Management
 - 5. Risk Management
 - 6. Quantitative Project Management
 - 7. Engineering Process Category
 - 8. Engineering Support Practice Category
 - 9. Process Management Practice Category
 - I.** Point 6—Better Economies by Delaying on CMM
 - J.** Achieving Maturity Levels 4 and 5



Agile Data Warehousing Project Management With SCRUM HDT823

VIII. Managing Adversity

- A. Gaining the Support of IT Managers**
 - 1. Why Agile Might Vex IT Managers
 - 2. Marketing Agile Approaches in General
 - 3. Agile Is Better In Terms of “Chaos”
 - 4. Agile Is Better In Terms of Opportunity Costs
 - 5. Agile Offers Better ROI
 - 6. Marketing ADW In Particular
 - i. Be More Than Code-and-Fix
 - ii. Ensure the Continuation of IT Services
 - iii. Support Common Architectures
 - iv. Provide Budgeting and Progress Metrics
 - v. Offer Visibility and Correction of Underperformance
 - vi. Ensure Application Quality
 - vii. Enhance IT Departmental Cohesion
 - 7. Possible Downsides Need to be Clear
 - i. Agile Team Members Cannot Be Easily Redeployed
 - ii. Must Wait a Bit Longer for Budgeting and Planning Estimates
 - iii. Must Keep the Product Owner Embedded in the Team
- B. Adversity From the Project Management Office**
 - 1. Mitigating Conflict with the PMO
- C. Adversity From IT Functional Groups**
 - 1. Mitigating Conflict with IT Groups
 - i. Guarding Against Sloppy Code
 - ii. Supporting Enterprise Data Integration
- D. Adversity From IT Service Vendors**
- E. Adversity From Business Stakeholders**
- F. Adversity From Teammates**
 - 1. Mitigating Adversity from Team Members