Lean & Agile Project Management for Large Programs & Projects

Dr. David F. Rico, PMP, CSM

Website: http://davidfrico.com
LinkedIn: http://www.linkedin.com/in/davidfrico
Facebook: http://www.facebook.com/profile.php?id=1540017424
Agenda

🌟 Overview of Agile Project Mgt.

- Intro to Agile Project Mgt.
- Types of Agile Project Mgt.
- Phases of Agile Project Mgt.
- Scaling Agile Project Mgt.
- Metrics for Agile Project Mgt.
- Cases of Agile Project Mgt.
- Summary of Agile Project Mgt.
Author

- DoD contractor with 27+ years of IT experience
- Large gov’t projects in U.S., Far/Mid-East, & Europe

- Published six books & numerous journal articles
- Adjunct at George Washington, UMUC, & Argosy
- Agile Program Management & Lean Development
- Expertise in metrics, models, & cost engineering
- Six Sigma, CMMI, ISO 9001, DoDAF & DoD 5000
What is Agility?

- **A-gil-i-ty** (ə-ˈji-lə-tē): Quickness, lightness, and ease of movement; To be very nimble:
  - The ability to create and respond to change in order to profit in a turbulent global business environment
  - The ability to quickly reprioritize use of resources when requirements, technology, and knowledge shift
  - A very fast response to sudden market changes and emerging threats by intensive customer interaction
  - Use of evolutionary, incremental, and iterative delivery to converge on an optimal customer solution
  - Maximizing the business value with right-sized, just-enough, and just-in-time processes and documentation
What are Agile Methods?

- **Adaptable** system development methodologies
- **Human-centric** method for creating business value
- **Alternative** to large document-based methodologies

### Agile Methods ‘Values’

<table>
<thead>
<tr>
<th>Agile Methods ‘Values’</th>
<th>Agile Methods ‘Principles’</th>
<th>Traditional Methods ‘Values’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Collaboration</td>
<td>Customer Interaction</td>
<td>Contract Negotiation</td>
</tr>
<tr>
<td>Individuals &amp; Interactions</td>
<td>High-Performance Teams</td>
<td>Processes &amp; Tools</td>
</tr>
<tr>
<td>Working System</td>
<td>Iterative Development</td>
<td>Comprehensive Documentation</td>
</tr>
<tr>
<td>Responding to Change</td>
<td>Adaptability or Flexibility</td>
<td>Following a Plan</td>
</tr>
</tbody>
</table>

How do Lean & Agile Intersect?

- Lean thinking provides the **what** (requirements)
- Agile thinking provides the **how** (implementation)
- Agile Methods are lean, light, adaptable, and flexible

<table>
<thead>
<tr>
<th>Agile Pillars</th>
<th>Agile Principles</th>
<th>Lean Pillars</th>
<th>Lean Principles</th>
<th>Other Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer collaboration</td>
<td>Intensive customer collaboration and interaction</td>
<td>Respect for people</td>
<td>Customer defines value</td>
<td>Economic view</td>
</tr>
<tr>
<td>Individuals and interactions</td>
<td>Small empowered high-performance multi-disciplinary teams</td>
<td></td>
<td>Customer pulls value</td>
<td>Fast feedback</td>
</tr>
<tr>
<td>Working systems and software</td>
<td>Iterative development of working operational systems and software</td>
<td>Continuous improvement</td>
<td>Continuous flow</td>
<td>Reduce batch size</td>
</tr>
<tr>
<td>Responding to change</td>
<td>Responding to change with flexible culture, process, and product</td>
<td></td>
<td>Continuous improvement</td>
<td>Control cadence</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Map value stream (eliminate waste)</td>
<td>Manage queue size</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Exploit variability</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Manage work-in-process</td>
</tr>
</tbody>
</table>

Essence of Agile Methods

- High degree of customer & developer interaction
- Highly-skilled teams producing frequent iterations
- Right-sized, just-enough, and just-in-time process

When to use Agile Methods

- On exploratory or research/development projects
- When fast customer responsiveness is paramount
- In organizations that are highly-innovative & creative


Agenda

Overview of Agile Project Mgt.

Intro to Agile Project Mgt.

Types of Agile Project Mgt.

Phases of Agile Project Mgt.

Scaling Agile Project Mgt.

Metrics for Agile Project Mgt.

Cases of Agile Project Mgt.

Summary of Agile Project Mgt.
“Agility” has many dimensions other than software.
Ranges from organizational to technological agility.
The focus of this brief is project management agility.
Today’s Environment

- Highly-unstable global and domestic markets
- Technology is evolving at an exponential speed
- Project plans cannot cope with this level of volatility

Need for a New Model

- Need for a **new model** of project management
- Cope with high-level of **uncertainty** and **ambiguity**
- With just the right balance of **flexibility** and **discipline**

Agile Project Management

- **APM** (ā-pē-ēm): Lightweight, flexible, adaptive, and collaborative; To be market or customer-responsive:
  - Rapidly and reliably creating value by engaging customers, continuously learning, and adapting
  - Sound, yet flexible process to manage projects under uncertainty, urgency, and a need for unique expertise
  - Managing the flow of human thoughts, emotions, and interactions in a way that produces business value
  - Values, principles, and practices to help project teams in coming to grips with a challenging environment

Values of APM

- Agile Manifesto (2001) focuses on collaboration
- DOI (2005) focuses on creating business value
- APM Values (2010) focus on all-around agility

**Agile Manifesto**
- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

**Declaration of Interdependence**
- Increase return on investment
- Deliver reliable results
- Expect uncertainty
- Unleash creativity and innovation
- Boost performance
- Improve effectiveness and reliability

**APM Values**
- Delivering value over meeting constraints
- Leading the team over managing tasks
- Adapting to change over conforming to plans

Agenda

Overview of Agile Project Mgt.
Intro to Agile Project Mgt.

» Types of Agile Project Mgt.

Phases of Agile Project Mgt.
Scaling Agile Project Mgt.
Metrics for Agile Project Mgt.
Cases of Agile Project Mgt.
Summary of Agile Project Mgt.
Scrum Project Management

- Created by Jeff Sutherland at Easel in 1993
- Product backlog comprised of customer needs
- Barely-sufficient project management framework

XP Project Management

- Created by Kent Beck at Chrysler in 1998
- Release plan is comprised of customer needs
- Lightweight, rigorous near-term planning element

**Release Planning**
- **Exploration Phase**
  - Build a Team
  - Write User Stories
  - Estimate User Stories
  - Split User Stories
  - Spike User Stories
  - Write User Tests

- **Commitment Phase**
  - Sort by Value
  - Sort by Risk
  - Set Velocity
  - Choose a Scope
  - Set Iteration Length
  - Develop Release Plan
  - Accept Tasks
  - Set Individual Velocity
  - Estimate Tasks
  - Analyze Schedules
  - Set Load Factors
  - Balance Tasks
  - New Release Plan
  - Select Tools
  - Adjust Teams

- **Steering Phase**
  - Select Iteration
  - Adjust Velocity
  - Insert New Stories
  - Select Partner
  - Write Unit Tests
  - Design and Code
  - Unit/Integration Test
  - User Acceptance Test
  - Record Progress

Flexible Project Management

- Created by Doug DeCarlo at Cutter in 2004
- Focus is on collaboration, scoping, and speed
- Thinner traditional project management approach

Adaptive Project Management

- Created by Sanjiv Augustine at CC Pace in 2005
- Builds agile cultures, mind-sets, and environments
- Leadership model for managing agile project teams

Agile Project Management

- Created by Jim Highsmith at Cutter in 2003
- Focus on strategic plans and capability analysis
- Most holistic agile project management framework

Innovation Lifecycle

- **Envision**
  - Product Vision
  - Product Architecture
  - Project Objectives
  - Project Community
  - Delivery Approach

- **Speculate**
  - Gather Requirements
  - Product Backlog
  - Release Planning
  - Risk Planning
  - Cost Estimation

- **Explore**
  - Iteration Management
  - Technical Practices
  - Team Development
  - Team Decisions
  - Collaboration

- **Launch**
  - Final Review
  - Final Acceptance
  - Final QA
  - Final Documentation
  - Final Deployment

- **Close**
  - Clean Up Open Items
  - Support Material
  - Final Retrospective
  - Final Reports
  - Project Celebration

Iterative Delivery

- **Technical Planning**
  - Story Analysis
  - Task Development
  - Task Estimation
  - Task Splitting
  - Task Planning

- **Development, Test, and Evaluation**
  - Development Pairing
  - Unit Test Development
  - Simple Designs
  - Coding and Refactoring
  - Unit and Component Testing

- **Operational Testing**
  - Integration Testing
  - System Testing
  - Operational Testing
  - Usability Testing
  - Acceptance Testing

- **Adapt**
  - Focus Groups
  - Technical Reviews
  - Team Evaluations
  - Project Reporting
  - Adaptive Action

Continuous

- Standups, Architecture, Design, Build, Integration, Documentation, Change, Migration, and Integration

Story Deployment

Agenda

Overview of Agile Project Mgt.
Intro to Agile Project Mgt.
Types of Agile Project Mgt.

Phases of Agile Project Mgt.

Scaling Agile Project Mgt.
Metrics for Agile Project Mgt.
Cases of Agile Project Mgt.
Summary of Agile Project Mgt.
Envision Phase

- Determine product vision and project objectives
- Identifies project community and project team
- The major output is a “Product Vision Box”

Diagram:

- **Delivery Approach**
  - Self-Organization Strategy
  - Collaboration Strategy
  - Communication Strategy
  - Process Framework Tailoring
  - Practice Selection and Tailoring

- **Product Vision**
  - Product Vision Box
  - Elevator Test Statement
  - Product Roadmap
  - Product Features
  - Product Vision Document

- **Product Architecture**
  - Product Skeleton Architecture
  - Hardware Feature Breakdown
  - Software Feature Breakdown
  - Organizational Structure
  - Guiding Principles

- **Project Community**
  - Get the Right People
  - Participant Identification
  - Types of Stakeholders
  - List of Stakeholders
  - Customer-Developer Interaction

- **Project Objectives**
  - Project Data Sheet
  - Key Business Objectives
  - Tradeoff Matrix
  - Exploration Factor
  - Requirements Variability

---

Speculate Phase

- Determine organizational capability/mission needs
- Identifies feature-sets and system requirements
- The major output is a “System Release Plan”

Explore Phase

- Determine technical iteration objectives/approaches
- Identifies technical tasks and technical practices
- The major output is an “Operational Element”

Adapt Phase

- Determine the effectiveness of operational elements
- Identifies customer feedback and corrective actions
- The major output is a “Process Improvement Plan”

Close Phase

- Determine project outcome and effectiveness
- Identifies strengths, weaknesses, and rewards
- The major output is a “Lessons-Learned Report”

Overview of Agile Project Mgt.
Intro to Agile Project Mgt.
Types of Agile Project Mgt.
Phases of Agile Project Mgt.

Scaling Agile Project Mgt.
Metrics for Agile Project Mgt.
Cases of Agile Project Mgt.
Summary of Agile Project Mgt.
Multi-Level Teams

- Enables projects to plan for the future and present
- Decomposes capabilities into implementable pieces
- Unclogs the drainpipes to let the execution flow freely

Multi-Level Planning

- Enables multiple level enterprise plans to co-exist
- Allows stakeholders to build viewpoint-specific plans
- Ensures capabilities are delivered at regular intervals

Multi-Level Backlog

- Enables multiple levels of abstraction to co-exist
- Allows customers and developers to communicate
- Makes optimum use of people’s time and resources

**Capability**
- Mission goal or objective level
- High-level business or product function
- Also called an Epic, i.e., multiple feature sets
- Comprises 18-90 days worth of work

**Feature Set**
- Cross-functional mission threads
- Related user stories that are grouped together
- Also called a Theme, i.e., implemented as an entity
- Comprises 6 to 30 days worth of work

**User Story**
- Functional, system-level requirements
- Simple requirement written by customer or user
- A small unit of functionality having business value
- Comprises 2 to 10 days worth of work

Multi-Level Coordination

- Enables lean and agile methods to scale-up
- Allows enterprises to create large-scale programs
- Unleashes optimum productivity and overall control

Multi-Level Governance

- Enables enterprises to achieve functional needs
- Allows programs to coordinate functional activities
- Ensures optimal technical performance is achieved

Agenda

Overview of Agile Project Mgt.
Intro to Agile Project Mgt.
Types of Agile Project Mgt.
Phases of Agile Project Mgt.
Scaling Agile Project Mgt.

Metrics for Agile Project Mgt.

Cases of Agile Project Mgt.
Summary of Agile Project Mgt.
Basic Agile Metrics

- Agile methods are based on traditional measures
- Size, effort, and velocity metrics are most common
- Top-notch shops use complexity and testing metrics

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Story, Story Point, Task, Function Point, LOC, etc.</td>
</tr>
<tr>
<td>Effort</td>
<td>Ideal or Actual Hours, Days, Weeks, Months, Years, etc.</td>
</tr>
<tr>
<td>Velocity</td>
<td>Story, Story Points, Function Points, or LOC per Iteration/Sprint</td>
</tr>
<tr>
<td>Complexity</td>
<td>McCabe, Halstead, Object-Oriented, Relational Database, etc.</td>
</tr>
<tr>
<td>Quality</td>
<td>Defect Density, Defect Removal Efficiency, Rayleigh, etc.</td>
</tr>
<tr>
<td>Testing</td>
<td>Tests Passed/Failed/Broken, Running Tested Features, etc.</td>
</tr>
<tr>
<td>Reliability</td>
<td>Mean Time to Failure, Mean Time between Failure, etc.</td>
</tr>
</tbody>
</table>

Burndown/Burnup Metrics

- Time expended is used for project tracking
- Tracked on a per-iteration or per-sprint basis
- Often described as a basic earned-value metric

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal Days</td>
<td>How many days something takes without interruptions</td>
</tr>
<tr>
<td>Actual Days</td>
<td>How many days something takes with interruptions</td>
</tr>
<tr>
<td>Ideal Hours</td>
<td>How many hours something takes without interruptions</td>
</tr>
<tr>
<td>Actual Hours</td>
<td>How many hours something takes with interruptions</td>
</tr>
<tr>
<td>User Stories</td>
<td>How many customer requirements have been satisfied</td>
</tr>
<tr>
<td>Story Points</td>
<td>How many units of software size have been satisfied</td>
</tr>
<tr>
<td>Technical Tasks</td>
<td>How many technical tasks have been completed</td>
</tr>
</tbody>
</table>

## Agile Cost Models

- **Costs** based on **productivity** and **quality** models
- Development costs based on $\text{LOC} \div \text{productivity rate}$
- Maintenance costs based on $\text{defects} \times 100 \times \text{KLOC} \times \text{MH}$

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Form</td>
<td>$(\text{LOC} \div \text{Productivity} + \text{Quality} \times 100) \times \text{Hourly Rate}$</td>
</tr>
<tr>
<td>XP</td>
<td>$(\text{LOC} \div 16.1575 + 0.7466 \times 100) \times \text{Hourly Rate}$</td>
</tr>
<tr>
<td>TDD</td>
<td>$(\text{LOC} \div 29.2800 + 2.1550 \times 100) \times \text{Hourly Rate}$</td>
</tr>
<tr>
<td>PP</td>
<td>$(\text{LOC} \div 33.4044 + 2.3550 \times 100) \times \text{Hourly Rate}$</td>
</tr>
<tr>
<td>Scrum</td>
<td>$(\text{LOC} \div 05.4436 + 3.9450 \times 100) \times \text{Hourly Rate}$</td>
</tr>
<tr>
<td>Agile</td>
<td>$(\text{LOC} \div 21.2374 + 1.7972 \times 100) \times \text{Hourly Rate}$</td>
</tr>
</tbody>
</table>

A major principle of Agile Methods is creating value. ROI is the measure of value within Agile Methods. There are seven closely related ROI measures:

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs</td>
<td>Total amount of money spent on agile methods</td>
</tr>
<tr>
<td>Benefits</td>
<td>Total amount of money gained from using agile methods</td>
</tr>
<tr>
<td>Breakeven</td>
<td>Point when the benefits of using agile methods exceed the costs</td>
</tr>
<tr>
<td>B/CR</td>
<td>Ratio of agile methods benefits to costs of using agile methods</td>
</tr>
<tr>
<td>ROI</td>
<td>Ratio of adjusted agile methods benefits to costs of using them</td>
</tr>
<tr>
<td>NPV</td>
<td>Present value of agile methods benefits that result from their use</td>
</tr>
<tr>
<td>Real Options</td>
<td>Value gained from incremental investments in high-risk projects</td>
</tr>
</tbody>
</table>
Agile EVM

- EVM has been adapted to Agile Methods
- EVM based on notion that total scope is known
- EVM is “not” well-suited for large-scale agile projects

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMB</td>
<td>Total number of story points planned for a release</td>
</tr>
<tr>
<td>SBL</td>
<td>Total number of iterations multiplied by iteration length</td>
</tr>
<tr>
<td>BAC</td>
<td>The planned budget for the release</td>
</tr>
<tr>
<td>PPC</td>
<td>Number of current iterations divided by planned iterations</td>
</tr>
<tr>
<td>APC</td>
<td>Total story points completed divided by story points planned</td>
</tr>
<tr>
<td>SPC</td>
<td>Story points of work completed from backlog during iteration</td>
</tr>
<tr>
<td>SPA</td>
<td>Story points added/subtracted from backlog during iteration</td>
</tr>
</tbody>
</table>

Overview of Agile Project Mgt.
Intro to Agile Project Mgt.
Types of Agile Project Mgt.
Phases of Agile Project Mgt.
Scaling Agile Project Mgt.
Metrics for Agile Project Mgt.

Cases of Agile Project Mgt.

Summary of Agile Project Mgt.
E-Commerce—Google

- Google started using agile methods in 2005
- Used it on one of their most profitable products
- Incrementally adopted agile one practice at a time

<table>
<thead>
<tr>
<th>Project Name</th>
<th>AdWords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Type</td>
<td>Pay-per-Click (PPC) Internet Advertising Mechanism</td>
</tr>
<tr>
<td>Project Size</td>
<td>20 teams of 140 people distributed over 5 countries</td>
</tr>
<tr>
<td>Product Size</td>
<td>1,838 user stories, 6,250 function points, 500,000 lines of code</td>
</tr>
<tr>
<td>Environment</td>
<td>Entrepreneurial, egalitarian, dynamic, unpredictable, informal, unstructured</td>
</tr>
<tr>
<td>Before APM</td>
<td>Chronic schedule delays, poor quality, unpredictability, poor estimation</td>
</tr>
<tr>
<td>APM Practices</td>
<td>Release planning, wikis for APM support, early testing and continuous integration</td>
</tr>
<tr>
<td>After APM</td>
<td>Better planning and estimates, earlier testing, better quality, large-scale adoption</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>Agile fit like a hand-in-glove, introduce agile methods slowly and then scale-up</td>
</tr>
</tbody>
</table>

Shrink-Wrapped—Primavera

- Primavera started using agile methods in 2004
- Used it on their flagship project management tools
- Adopted agile all-at-once with top-down mgt. support

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Primavera</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Type</td>
<td>Enterprise Project Management Tool</td>
</tr>
<tr>
<td>Project Size</td>
<td>15 teams of 90 people collocated at one site</td>
</tr>
<tr>
<td>Product Size</td>
<td>26,809 user stories, 91,146 function points, 7,291,666 lines of code</td>
</tr>
<tr>
<td>Environment</td>
<td>Top-down, hierarchical, command and control, traditional, waterfall approach</td>
</tr>
<tr>
<td>Before APM</td>
<td>Poor relationships, quality, usability, and customer satisfaction, functional silos, 18-hour days, 7-day work weeks, frustration, disappointment, apathy, exhaustion</td>
</tr>
<tr>
<td>APM Practices</td>
<td>Release planning, agile project management tools, automated testing tools</td>
</tr>
<tr>
<td>After APM</td>
<td>75% quality and 40% cycle time improvement, 40-hour work week, 0% attrition</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>Agile results in better communication, motivation, and empowerment</td>
</tr>
</tbody>
</table>

FDA suppliers started using agile methods in 2008
Used it on most stringent Class 3 certified products
Used to modernize 1990s era products & processes

<table>
<thead>
<tr>
<th>Project Name</th>
<th>m2000 Real-time PCR Diagnostics System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Type</td>
<td>Human Blood Analysis Tool (i.e., HIV-1, HBV, HCV, CT, NG, etc.)</td>
</tr>
<tr>
<td>Project Size</td>
<td>4 teams of 20 people collocated at one site</td>
</tr>
<tr>
<td>Product Size</td>
<td>1,659 user stories, 5,640 function points, 451,235 lines of code</td>
</tr>
<tr>
<td>Environment</td>
<td>FDA-regulated medical devices, real-time, safety-critical, Class III–most stringent</td>
</tr>
<tr>
<td>Before APM</td>
<td>Cumbersome process, poor quality, long cycle time, slow big-bang integration, obsolete, hard-to-staff tools and methods, inability to keep pace with changing requirements, intense market competition, exponential rate of technological change, fewer resources</td>
</tr>
<tr>
<td>APM Practices</td>
<td>Release planning, lighter-weight agile testing techniques, continuous integration</td>
</tr>
<tr>
<td>After APM</td>
<td>25% cycle time and staff-size reduction, 43% cost reduction, fewer defects</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>Agile enables the ability to balance fast cycle time with high-quality safety-critical solutions</td>
</tr>
</tbody>
</table>

Law Enforcement—FBI

- IC started using agile methods following 9/11
- Used it on billion dollar transformation initiatives
- Goal is to catch bad guys better, faster, and cheaper

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Inter-Agency Intelligence Sharing System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Type</td>
<td>Domestic Terrorist Database/Data Warehouse</td>
</tr>
<tr>
<td>Project Size</td>
<td>3 teams of 12 people collocated at one site</td>
</tr>
<tr>
<td>Product Size</td>
<td>643 user stories, 2,188 function points, 175,000 lines of code</td>
</tr>
<tr>
<td>Environment</td>
<td>CMMI Level 3, ISO 9001, government-mandated document-driven waterfall life cycle, emerging federal directives for more information sharing and integration among intelligence community partners, rapidly changing customer requirements</td>
</tr>
<tr>
<td>Before APM</td>
<td>Unresponsive waterfall life cycles, chronic schedule delays, anxious customers, unhappy developers, resource focus on becoming CMMI Level 3 certified caused everyone to lose track of the real goal, which was to “catch bad guys”</td>
</tr>
<tr>
<td>APM Practices</td>
<td>Release planning, user stories, test-driven development, continuous integration</td>
</tr>
<tr>
<td>After APM</td>
<td>50% quality improvement, 200% productivity increase, FBI created policy for agile methods</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>Agile enables fast response times, customer satisfaction, and ability to “catch bad guys”</td>
</tr>
</tbody>
</table>

U.S. DoD—STRATCOM

- U.S. DoD started using agile methods following 9/11
- Used it on billion-dollar software-intensive systems
- Goals are to respond to rapidly emerging threats

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Strategic Knowledge Integration Website (SKIweb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Type</td>
<td>Knowledge Management System (KMS)—Advanced Search Capability</td>
</tr>
<tr>
<td>Project Size</td>
<td>3 teams of 12 people collocated at one site</td>
</tr>
<tr>
<td>Product Size</td>
<td>390 user stories, 1,324 function points, 105,958 lines of code</td>
</tr>
<tr>
<td>Environment</td>
<td>Traditional linear documentation-based development, contract-oriented, hierarchical communication, rapidly changing operational requirements, need for leaner U.S. military force, seeking better and faster ways of getting critical information to decision makers, decentralization, migration to net-centric service oriented architectures, egalitarian decisions</td>
</tr>
<tr>
<td>Before APM</td>
<td>Long cycle times, dissatisfied customers, unresponsive life cycles, poor quality</td>
</tr>
<tr>
<td>APM Practices</td>
<td>Release planning, frequent customer collaboration, continuous integration</td>
</tr>
<tr>
<td>After APM</td>
<td>Good teamwork, 200% productivity increase, improved quality, fewer defects</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>Agile improves customer satisfaction/communication, and overall product quality</td>
</tr>
</tbody>
</table>

Overview of Agile Project Mgt.
Intro to Agile Project Mgt.
Types of Agile Project Mgt.
Phases of Agile Project Mgt.
Scaling Agile Project Mgt.
Metrics for Agile Project Mgt.
Cases of Agile Project Mgt.

Summary of Agile Project Mgt.
Advanced Agile Measures

- Agile Methods are a fundamentally new paradigm
- Agile Methods are “not” lighter Traditional Methods
- They should not be viewed through a traditional lens

Benefits of Agile Methods

- Analysis of 23 agile vs. 7,500 traditional projects
- Agile projects are 54% better than traditional ones
- Agile has lower costs (61%) and fewer defects (93%)

Myths about Agile Methods

- Common myths abound, although agile methods have been around for ~20 years:
  - Agile methods are only for software development
  - Agile methods are only for small co-located teams
  - Agile methods have no documentation
  - Agile methods have no requirements
  - Agile methods need traditional system architectures
  - Agile methods have no project management
  - Agile methods are undisciplined and unmeasurable
  - Systems built using agile methods are unmaintainable and insecure
Conclusions

- Traditional methods are well-suited for predictability
- Agile Methods are well-suited for high uncertainty
- It comes down to efficiency versus effectiveness

Traditional Project Management
- Predictable situations
- Low-technology projects
- Stable, slow-moving industries
- Low-levels of technological change
- Repeatable operations
- Low-rates of changing project performance
- Long-term, fixed-price production contracts
- Achieving concise economic efficiency goals
- Highly-administrative contracts
- Mass production and high-volume manufacturing
- Highly-predictable and stable market conditions
- Low-margin industries such as commodities
- Delivering value at the point-of-plan

Agile Project Management
- High-levels of uncertainty and unpredictability
- High-technology projects
- Fast-paced, highly-competitive industries
- Rapid pace of technological change
- Research-oriented, discovery projects
- Large-fluctuations in project performance
- Shorter-term, performance-based RDT&E contracts
- Achieving high-impact product/service effectiveness
- Highly-creative new product development contracts
- Customer-intensive, one-off product/service solutions
- Highly-volatile and unstable market conditions
- High-margin, intellectually-intensive industries
- Delivering value at the point-of-sale

New Book on Agile Methods

- Guide to Agile Methods for business leaders
- Communicates business value of Agile Methods
- Rosetta stone to Agile Methods for traditional folks

Table of Contents
1. Introduction to Agile Methods
2. Values of Agile Methods
3. History of Agile Methods
4. Antecedents of Agile Methods
5. Types of Agile Methods
6. Practices of Agile Methods
7. Agile Project Management
8. Agile Software Engineering
9. Agile Support Processes
10. Agile Tools and Technologies
11. Comparison of Agile Methods
12. Agile Metrics and Models
13. Surveys of Agile Methods
15. ROI Metrics of Agile Methods
16. Measures of Agile Methods
17. Costs of Agile Methods
18. Benefits of Agile Methods
19. ROI of Agile Methods
20. NPV of Agile Methods
21. Real Options of Agile Methods
22. Business Value of Agile Methods
23. Agile vs. Traditional Methods
24. Future of Agile Methods

http://davidfrico.com/agile-book.htm (Description)
http://www.amazon.com/dp/1604270314 (Amazon)