Case Study: Applying Agile Software Practices to Systems Engineering

Presented by:
Matthew R. Kennedy, PhD
IEEE Software Technology Conference
April 10th, 2013
Definitions

- **Agility**
  - “The speed of operations within an organization and speed in responding to customers (reduced cycle times)” (Mass. Inst. Tech.)

- **Systems Engineering**
  - “The overarching process that a program team applies to transition from a stated capability to an operationally effective and suitable system” (DAU)

- **Software Intensive System**
  - “A system in which software represents the largest segment in one or more of the following criteria: system development cost, system development risk, system functionality, or development time” (DAU)
IT Organizational Structure
Aspects of Developing a SIS

• Business Aspect
  – Responsible for the overall acquisition: contracting, funding, operational requirements, and system delivery structure

• System Aspect
  – Further decompose the requirements and allocate them to software or hardware. Overall technical management

• Software Aspect
  – Software items
Different Focus – Same Goal

Software Intensive Systems are held captive by the slowest Aspect
Agility within the Software Aspect

In the beginning…
Agile Manifesto

• Feb 2001 – Representatives from several software development* methodologies met to find a common ground on an alternative to documentation driven, heavyweight software development processes

• Established:
  – Core Values
  – Principles

*Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming

www.agilemanifesto.org
Agile Software Development Methodologies

- Scrum
- eXtreme Programming (XP)
- Dynamic Systems Development Method
- Rapid Application Development
- Crystal
- Kanban
- ...

Software Aspect
Software Aspect Frameworks

- Scrum
Agility within the Business Aspect
Fast, Inexpensive, Simple, Tiny (F.I.S.T.) Manifesto*

• Similar to the Agile Manifesto but targets the Business Aspect
• Values
  – Talent trumps process
  – Teamwork trumps paperwork
  – Leadership trumps management
  – Trust trumps oversight

F.I.S.T.

• Principles
  - A project leader’s influence is inversely proportional to the project’s budget and schedule;
  - Constraints foster creativity;
  - Fixed funding and floating requirements are better than fixed requirements and floating funding;
  - An optimal failure costs a little and teaches a lot;
  - Complexity is cost, Complexity reduces reliability, Simplicity scales, Complexity does not;
  - Iteration drives learning, discovery and efficiency.

• Implementation Guidelines
  - Minimize team size and maximize team talent;
  - Use schedules and budgets to constrain the design;
  - Insist on simplicity in organizations, processes and technology;
  - Incentivize and reward under runs;
  - Requirements must be achievable within short time horizons;
  - Designs must only include mature technologies;
  - Documents and meetings: have as many as necessary, as few as possible;
  - Delivering useful capabilities is the only measure of success.
Business Aspect Frameworks

- Business Capabilities Lifecycle Framework

  - Investment Management
    - Business Capability Definition
    - MDD
    - Up to 12 Months

  - Execution
    - Prototyping
      - Pre-ED Review
      - 12 Months*
    - Engineering Development
    - Limited Fielding
      - IOC
      - IOT&E
      - FDD
    - Full Deployment
    - Operations and Support
    - 18 Months* MS B to IOC

- Defense Science Board Agile Framework

  - Milestone Build Decision
  - CDD
  - RELEASE 1
    - Development & Demonstration
      - Iteration 1
      - Iteration 2
      - Iteration “N”
  - RELEASE 2
    - Prototypes
    - Development & Demonstration
      - Iteration 1
      - Iteration 2
      - Iteration “N”
  - RELEASE “N”
    - Prototypes
    - Development & Demonstration
      - Iteration 1
      - Iteration 2
      - Iteration “N”
  - Continuous Technology/Requirements Development & Maturation
Agility within the System Aspect
Going from the Business to the Software Aspect
• Agile practices from analysis of agile implementations in the Business and Software Aspects:

<table>
<thead>
<tr>
<th>Incremental Development</th>
<th>Small Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterative Development</td>
<td>Time Boxing</td>
</tr>
<tr>
<td>Short Time-lines</td>
<td>Lean Initiatives</td>
</tr>
<tr>
<td>Retrospectives (Lessons learned)</td>
<td>Prototyping</td>
</tr>
<tr>
<td>Empowered / Self-organizing / Managing teams</td>
<td>Continuous User Involvement</td>
</tr>
<tr>
<td>Prioritized Product Backlog (Requirements)</td>
<td>Co-located Teams</td>
</tr>
</tbody>
</table>

Kennedy / Ward

• And created an Agile Framework
Agile System Eng. Framework

1 RELEASE
1-N INCREMENTS

RELEASE PLANNING
Increment Planning
1-M Sprints
SPRINT
SPRINT
System Integration
Integration Retrospective
Increment Planning
1-M Sprints
SPRINT
SPRINT
Integration Retrospective
System Integration
Integration Retrospective
Increment Retrospective
Integration Retrospective
Integration Retrospective
Increment RETROSPECTIVE
RELEASE RETROSPECTIVE

DAU
System Aspect
Framework Phases

• Each phase has predefined input / exit criteria that fosters system engineering and agile practices
  – Release
  – Increment
  – Integration
  – Sprint

• Each phase ends with a retrospective
Sprint

• A Sprint could use:
  – Scrum
  – XP
  – Manufacturing Processes
  – Waterfall
Sprint Input / Exit Criteria

• **Input Criteria**
  - Sprint Backlog (work to be performed)
  - Specifications / Interfaces
  - Identification of Customer
  - Definition of “Done”

• **Activities in this Phase: Item Development**

• **Exit Criteria: Completed / user-accepted product(s)**
Managing Risk – “Black-box Trust”

- Input Criteria
- User accepted
- Time-boxed

- F.I.S.T.:
  - Constraints Foster Creativity
  - Trust Trumps Oversight
  - Delivering useful capabilities is the only measure of success

- Agile Manifesto:
  - Breaking big work down into smaller components that can be completed quickly.
  - Measuring progress by the amount of completed work

Input Criteria (constraints)
Exit Criteria

Time-Boxed

User Accepted Product
Aspects: Holistic View
Case Study

Overview
Company Background

- ISO 9001:2008 registered company
- Over a 100-year history
- Produces various Information Technology (IT) solutions to customers worldwide.
- Initially a manufacturing organization that has expanded to include:
  - Manufacturing,
  - Mechanical and
  - Software departments
Project Background

- The system under development is a major modification to an existing component
  - New functionality
  - Upgrades to hardware, firmware, software*, and manufacturing components
- Developed systems in the past with the same personnel
- $1.3M budget
- 13 months

* The software component has been using agile software development techniques
Project Complexity

- Roughly 50 percent of the design and mechanical components were outsourced to leverage emerging technologies
- Manufacturing facilities that are located both in and outside of the United States (US)
- Delivered product must conform and/or be certified in several specifications (ANSI, MIL-STD)
Defining the Problem

- Their products were becoming routinely late, over budget, and did not include the planned functionality
- Seeking to improve predictability
What do we measure?

• Metrics based on 12 previous projects completed from the company that were of similar size and scope to the project in the case study

• Estimated vs. Actuals
  – Time to completion
  – Functionality delivered
  – Cost

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Cost Difference from Estimate</td>
<td>+2.5%</td>
</tr>
<tr>
<td>Average Schedule Difference from Estimate</td>
<td>+30%</td>
</tr>
<tr>
<td>Average Functionality difference from planned</td>
<td>-5%</td>
</tr>
</tbody>
</table>
Major Project Milestones

- **Milestone 1**
  - Design, development, manufacturing, and passing internal QA

- **Milestone 2**
  - Release to a limited user base and external testing

- **Milestone 3**
  - Mass production and customer sales of the final product
Case Study

Execution
Delivery of Framework and Practices

- Delivered in two face-to-face sessions
- Quarterly and ad hoc teleconferences and virtual meetings to gather actual implementation data
- 75+ emails exchanged
Application of the Framework and Practices

• Upon acceptance of the practices and framework, their project plan and delivery strategy was reengineered.
• Used the frameworks defined input and exit criteria as well as the activities for each phase.
• Moved from a once through development methodology to:
  – Three Increments with each containing Sprints that were designed to coincide with key integration points in which a combination of internally and externally developed hardware, firmware, enclosures and / or software needed to be integrated and tested by the Quality Control (QC) group.
• Prioritized their development primarily based on risk.
Example Structure*

Details were removed so the company could not be identified. Input / Exit criteria contained more detail in practice.
• They did not deviate from their standard terminology where possible*
  – Used terms familiar to their organization such as Alpha1, Alpha2 versus iteration1 and iteration2

*This semantic difference had no effect on the framework structure but decreased the organizational learning curve to utilize the agile framework and practices
Product Results (Milestone 1)

• 70 Units were manufactured and successfully passed internal Quality Control (QC)
  – Schedule
    • Scheduled for 27.5 weeks but completed in 29 weeks
      – This was a 5.5 percent schedule overrun
  – Functionality
    • Delivered 100% of the planned functionality
  – Cost*
    • 5% under budget

*Because the cost was due to a vendor renegotiation the cost fluctuation was not due to the framework or practices utilized during product development. Without the renegotiation they would be <1% under their estimate
Compared Results

<table>
<thead>
<tr>
<th>Compared</th>
<th>Past Performance</th>
<th>New Model</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Difference from Estimate</td>
<td>+2.5%</td>
<td>-*5%</td>
<td>7.5% Improvement</td>
</tr>
<tr>
<td>Schedule Difference from Estimate</td>
<td>+30%</td>
<td>+5.5%</td>
<td>24.5% Improvement</td>
</tr>
<tr>
<td>Functionality difference from planned</td>
<td>-5%</td>
<td>0%</td>
<td>5% Improvement</td>
</tr>
</tbody>
</table>

*Because the cost was due to a vendor renegotiation the cost fluctuation was not due to the framework or practices utilized during product development. Without the renegotiation they would be <1% under their estimate.
Issues Encountered

• Scheduling priorities – Other projects took priority in manufacturing, testing, or development;
• Manning issues;
• Retesting of subsystems during development due to unsuccessful QC;
• Higher than average failure rates on third party circuit boards;
• Fluctuation in material costs;
• Manufacturing lines shut down;
• Delays in receiving ordered parts
Overall Case Study Feedback

• The benefits of the framework and principles were seen outside of the engineering division and stretched into marketing
  – Increase in predictability of delivery dates allowed the marketing division to better plan for the marketing aspect of the product

• Very little resistance to the change
  – Project manager noted that “all future projects will be run in a similar fashion”

• The initial phase review was completed early
  – This was noted but management as “first time in company history”
Case Study Observations

- Increased project tracking capability
- Ease of project flexibility
- In addition to Agile Practices, the Framework inherently fostered System Engineering Best Practices
  - Configuration Management
  - Interface Management
  - Risk Management
  - Technical Planning
Agility

- Implementing agility is a different masterpiece for each organization
  - Identify your puzzle and SOLVE IT!

“A problem well stated is a problem half-solved.” - Charles Kettering
Questions