Evolutionary Capabilities
Developed and Fielded in Nine Months

Rapid and Reliable Development, CrossTalk, May/June 2009

Authors: Ms. Portia Crowe, US ARMY, Program Executive C3T, PM-Battle Command
Dr. Robert Cloutier, Stevens Institute of Technology
**Evolutionary Capabilities Developed and Fielded in Nine Months (BRIEFING CHARTS)**

Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT.
Agenda

• Challenges
• Program Needs
• Solution Approach
• Emergent Findings
• Learned Tenets
• Success
• Conclusion
Program Needs

“Develop a readiness reporting system that most accurately reflects the status of the unit to accomplish the mission they are most likely to do.”

GEN Casey, CSA

Implement senior leader guidance
Support strategy and doctrine
Simplify the readiness reporting process
Maintain necessary readiness visibility of Army units

Community of Stakeholders to include HQDA G3/5/7 Office of Defense Readiness, TRADOC, FORSCOM, PM Battle Command-SBC
Challenges

• Rapid development while following acquisition directives
  – Legacy and new system/technologies interoperability
  – Close user and stakeholders collaboration
  – Change of thought from waterfall to agile

• Complex environments

• Changing and unforeseen requirements
Challenges (continued)

• Develop an agile and adaptable system to support new technologies and requirements
  • Unit Status Reporting Tool

• Agile program management to include risk management

• Rapidly field capabilities
  – Buy in of leadership
  – Training users for new capabilities

Migrate a legacy, hardware dependant, client-server architecture to a web-based service enabled, hardware independent, secure environment (SIPRnet)
Transition from Legacy to Modern

Legacy business processes not able to handle new requirements
Existing capabilities
No robust failover strategy
Old hardware
Standalone Application
Stove-piped data
Swivel chair operations

PC ASORTS

No operational downtime
Trained Force
Multiple Contractors

IN 9 MONTHS!!

Unforeseen requirements
Flexibility of System & Process for emerging capabilities
New Architecture with robust failover and HW
Single Sign-on for suite of applications on one portal
Relationship between apps
Web based

Defense Readiness Reporting System-Army (DRRS-A)
Solutions Approach

• Agile Methodology (SCRUM)
  • Work off of a solution *backlog of requirements* which serves as a record of your requirements
    • Backlog is prioritized to ensure your most critical items are addressed first
  • Average 30 day Sprints (but could be flexible in length)
    • Development team commits to stories from the backlog
    • Work collaboratively with customer to develop stories
    • *Sprint Review* at end of sprint to review results with customer, get feedback and adapt or re-prioritize backlog stories as necessary
    • Delivery of *working software* at end of each sprint for hands-on use from a test server

• Full Dress Rehearsal Test Strategy
  • Done before each major delivery
  • Deliveries approximately every 60 days
  • User input on usability, effectiveness, quality
Scrum Development Process

Scrum: 15 minute daily meeting
Team members respond to basic questions:
1) What did you do since last Scrum meeting
2) Do you have any obstacles?
3) What will you do before the next meeting?

Product Backlog
Prioritized features desired by customer

Sprint Backlog
Features assigned to Sprint
Backlog items added by team members

Sprint

Working Increment of the software

24 h

30 days
Took an Overall Agile Approach to Systems Engineering

- Agile approach to ensure customer and user needs were met in a timely manner
- Evolvement of linear life cycle approach to working life cycle phases in parallel for rapidness
- Monthly requirements (new, enhancements, changes) and prioritization
- Training users and training materials were done early and often
- Weekly IPT meetings + Scrum sessions = constant collaboration and coordination
- Operational test events were conducted at coordinated checkpoints with bug/fix scheduled as a story in the backlog (3 events was magic number before fielding)
- Strong dedicated team (doesn’t have to be co-located) with team ownership of product
# Emergent Agile Culture Changes

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liberty to be dynamic</td>
<td>Agility needs dynamic processes while adhering to acquisition milestones</td>
</tr>
<tr>
<td>Non-linear; Cyclical and non-sequential</td>
<td>Life cycle behavior not like traditional waterfall models or linear frameworks; decreasing cycle times</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Conform to changes such as capability and environment</td>
</tr>
<tr>
<td>Simultaneous development of phase components</td>
<td>Rapid fielding time may not lend to traditional phase containment (i.e. training and SW development together)</td>
</tr>
<tr>
<td>Ease of Change</td>
<td>Culture shift to support change neutrality; ease of modification built into architecture and design</td>
</tr>
<tr>
<td>Short Iterations</td>
<td>Prototyping, demonstrating and testing can be done in short iterative cycles with tight user feedback loop</td>
</tr>
<tr>
<td>Light-weight phase attributes</td>
<td>Heavy process reduction such as milestone reviews, demonstrations, and risk management</td>
</tr>
</tbody>
</table>
Learned Tenets

- New functionality should have clear requirements and close functional proponent involvement to achieve “develop once and leave it” paradigm
- Start security certification process as soon as feasible
- Foster good relationships with stakeholders to encourage collaboration and ownership
- Involve users of new capabilities and functionality as change is usually unwanted but necessary
- Risk management is an oxymoron in agile methodologies. It can, and must become an inherent part of any agile approach
- Plan, communicate and help one another during this process—it’s a team effort
Success

- Successfully fielded DRRS-A in 9 months with:
  - A trained Force (Close to 5500 users)
    - train the trainer
    - application training
    - interactive media
    - user guides for self-teaching, and site training)
  - Assessable, secure and easy to use applications
  - Operationally effective and accurate - unit status reporting went from 82% a month to 98% reporting units
- Architecture is more robust (implements a service-oriented architecture)
- Agile methodology resulted in smaller, close knit development team
- Lifecycle cost savings exceeds over $2M and rising across Army
- Replicated success by developing the DRRS-Marine Corp applications in 8 months
More Success

• Implemented net-centric capabilities
  – Data is now understandable (data tagging), sharable and visible
  – Web services registered in NCES Registry

• Collaboration of multiple contractors (i.e. Lockheed Martin and Accenture)

• Continuing to rapidly deliver new and enhanced capabilities to the field
Conclusion

• Commercial best practices such as agile engineering can be used in DoD
  – However, it requires strong leadership
  – Commitment from the top
  – Organizational culture changes

• Initial intent was to use the agile approach for software only, however, quickly extended that to all systems engineering functions on DRRS-A
  – This became a critical success factor for the project