Successful Integration of Agile Development Techniques within DISA

Sean Cohan, Michael Gallaher, Chris Eldredge
The Customer

- The Customer

Defense Information Systems Agency
Department of Defense

Solutions at the Speed of Change

- Responsible for planning, engineering, acquiring, fielding, and supporting global net-centric solutions to serve the needs of the President, Vice President, the Secretary of Defense, and other DoD Components
The Program: JOPES

A combination of joint policies and procedures, supported by IT, designed to provide joint commanders and planners with a capability to plan and conduct joint military operations

DISA is the project sponsor
Who Uses JOPES?

The Joint Planning and Execution Community (JPEC)

The JPEC consists of headquarters, commands, and agencies involved in the training, preparation, movement, reception, employment, support, and sustainment of military forces assigned or committed to a theater of operations or objective area. It typically consists of the Joint Staff, Services, Service major commands, unified commands, subunified commands, transportation component commands, joint task forces (as applicable), Defense Logistics Agency, and other Defense agencies as may be appropriate to a given situation. The JPEC is involved in both deliberate and time-sensitive planning.
What Does JOPES Do?

- Deploying Forces from Origin to Destination
- Identification of Units
- Quantifying Lift Requirements
  - Equipment
  - Passengers
- Validation of Requirements to USTC
- Scheduling & Movement
  - Airlift & Sealift Missions
The Net-Centric Objective: The Problem

End-User Consumer
“What data exists?”
“How do I access the data?”
“How do I know this data is what I need?”
“How can I tell someone what data I need?”

End-User Producer
“How do I share my data with others?”
“How do I describe my data so others can understand it?”

Organization “A”
User is unaware this data exists

Organization “B”
User knows this data exists but cannot access it because of organizational and/or technical barriers

Organization “C”
User knows data exists and can access it but may not know how to make use of it due to lack of understanding of what data represents
The Net Centric Objective: The Solution - Service Oriented Architecture

**Service Producer**
Data and applications available for use, accessible via services. Metadata added to services based on producer's format.

- **Describes** content using metadata
- **Posts** metadata in catalogs for discovery
- **Exposes** data and applications as services

**Service Consumer**
Automated search of data services using metadata. Pulls data of interest. Based on producer registered format and definitions, translates into needed structure.

- **Searches** metadata catalogs to find data services
- **Analyzes** metadata search results found
- **Pulls** selected data based on metadata understanding

---

**Publish**

**Discover**

**Invoke**

**Enabled Inf**

- Service Registries
- Messaging Services
- Data Services
- Transformation Services
The Agile Objective

- Establish Automated Testing Development
- Establish Automated Build Environment

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Build</th>
<th>Revision</th>
<th>Tag</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOA-service</td>
<td>44 Minute(s)</td>
<td>1334</td>
<td>r68579</td>
<td>trunk</td>
<td>PASS</td>
</tr>
<tr>
<td>ACOA-test-utils</td>
<td>49 Minute(s)</td>
<td>40</td>
<td>r68579</td>
<td>trunk</td>
<td>PASS</td>
</tr>
<tr>
<td>ACOA-ui</td>
<td>40 Minute(s)</td>
<td>1623</td>
<td>r68586</td>
<td>trunk</td>
<td>FAIL</td>
</tr>
<tr>
<td>WebJET-api</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BUILDING</td>
</tr>
<tr>
<td>WebJET-war</td>
<td>13 Minute(s)</td>
<td>289</td>
<td>r68576</td>
<td>trunk</td>
<td>PASS</td>
</tr>
</tbody>
</table>
The Agile Beginning

- The Infrastructure
  - Wiki
  - Version Control System – Subversion
  - Continuous Integration Tool – AntHill
  - Open Area
  - Pairing Stations
  - White Boards
  - Cork Boards
  - Two-week iterations
  - Java
  - Struts
  - Axis
The Importance of the Coach

- Putting the Right Infrastructure in Place
- Educating Developers (reading lists, brown-bags)
- Showing how TDD is done
- Explaining Stories and Tasks
- Explaining how to Estimate Stories and Tasks
- Incorporating Stand-up Meetings
- Showing how to Mercilessly Refactor code
- Showing how to Continuously Integrate
- Showing how to Pair Program
- Keeping Developers from Reverting into Non-Agility
Improved Continuous Integration Tool

- Vulcan - (http://code.google.com/p/vulcan)
  - Initial version addressed continuous integration capabilities
  - Subsequent versions address build & release workflow
  - Adopted as portal for both continuously integrated builds and release candidates
Improved Continuous Integration Tool

- Import project from repository in one step

  **Import Project**

  - **Project URL:** http://svn.apache.org/repos/asf/jakarta/commons/proper/io/trunk/pom.xml
  - **Child Projects**
    - Create a single project.
    - Create a separate project for each child project.
  - **Project Name Conflicts**
    - Cancel and show an error message.
    - Use existing project without modifying it.
    - Overwrite the existing project.
  - **BuildSchedulers**
    - All Hours
    - Business Hours
    - Nightly
  - **Import**

- Build from release tag

- Test metrics including test failures

**Test Failures**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Failing Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>ant.AntBuildToolInvocationTest.testAntSeesRevisionAndTagName</td>
<td>110</td>
</tr>
<tr>
<td>ant.AntBuildToolInvocationTest.testAntSeesRevisionAndTagNameWithDifferentNames</td>
<td>110</td>
</tr>
<tr>
<td>maven.Maven2BuildToolInvocationTest.testCompileFails</td>
<td>New</td>
</tr>
<tr>
<td>maven.Maven2BuildToolInvocationTest.testGetsEvents</td>
<td>New</td>
</tr>
<tr>
<td>maven.MavenBuildToolInvocationTest.testGetsEvents</td>
<td>New</td>
</tr>
<tr>
<td>maven.MavenBuildToolInvocationTest.testHasRevisionAndLabelDefined</td>
<td>New</td>
</tr>
</tbody>
</table>
Management Support and Time to Mature

- Granted Adequate Time for Agile Team to Prove Themselves
  - Provided Open Area
  - Allowed Paired Programming
  - Provided Pairing Stations
  - Provided White Boards/Cork Boards
  - Allowed Team to Learn New Development Methodology
  - Allowed Team to Learn New Tool
Support from our DISA Customer

- Demos of Agile Development Techniques

<table>
<thead>
<tr>
<th>Project/Build Status</th>
<th># Classes</th>
<th>Line Coverage</th>
<th>Branch Coverage</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOM-all</td>
<td>3</td>
<td>95%</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td>ACOM-asp</td>
<td>3</td>
<td>85%</td>
<td>83%</td>
<td>1.214</td>
</tr>
<tr>
<td>ACOM-asp</td>
<td>4</td>
<td>82%</td>
<td>100%</td>
<td>0.0</td>
</tr>
<tr>
<td>ACOM-asp</td>
<td>21</td>
<td>77%</td>
<td>100%</td>
<td>1.222</td>
</tr>
<tr>
<td>ACOM-asp</td>
<td>12</td>
<td>85%</td>
<td>64%</td>
<td>0</td>
</tr>
<tr>
<td>ACOM-asp</td>
<td>4</td>
<td>72%</td>
<td>98%</td>
<td>0</td>
</tr>
<tr>
<td>ACOM-asp</td>
<td>8</td>
<td>93%</td>
<td>97%</td>
<td>2.417</td>
</tr>
</tbody>
</table>
DISA Moves Towards Agility

- The new DISA Director wants to change the time it took to get a release of software into production.
- DISA Test Engineering (TE) Branch Chief Begins to Implement Agile Testing.
- TE testers are working to become integrated with our Pragmatics testers earlier in the test phase.
- Our DISA Customer Decided to Leverage our Three-Month Internal Release Cycle.
- DISA is establishing the Federation Development Certification Environment (FDCE).
Successful Deliveries

- March 2006
  - Two agile applications put in production in the fall of 2006

- March 2007
  - Nine more agile applications delivered; Scheduled for production in fall of 2007
  - First three-month release delivered in June 2007
  - Delivering additional three-month releases in late August and November of 2007

- Customer confidence has increased resulting in increased business
  - New agile application developed and delivered from scratch in 6 weeks.
Challenges

- Lack of Onsite Customer/End User
- Integration of Non-Agile Developers
- Endless Iterations
- Integration of IV&V Group
Lessons Learned

- Get a Good Experienced Coach
- Get Support and Buy-in From Management
- A Customer aligned with Agile Principles (e.g., shorter, quicker releases, automated testing and deployments) enables/accelerates successful Agile Development implementation