AGILE DEVELOPMENT

U.S. POSTAL SERVICE

FEBRUARY 19, 2015
Agenda

1. State of Software Projects
2. USPS Overview
3. Agile Program
4. Agile Engineering
5. Scaled Agile Framework
6. USPS Retrospective
7. What’s Next for the USPS
Industry Situation

- 43%
- 39%
- 18%
- 74%
- 59%
Industry Metrics

What drove us here – industry-wide view

- **43%**
  - Challenged: Late, Over Budget, or Missing Key Features
  - 74% - schedule overruns
  - 59% - cost overruns

- **18%**
  - Failed: Cancelled or Never Used

- **39%**
  - Successful, but
    - 20% - features used
    - 50% - features hardly ever used (debatable value)

*Source: Chaos Report (2012).*
Scope of USPS IT Support

✓ One of the Largest Computing Infrastructures in the World
✓ 1200 IT Employees
✓ 894 Business Applications Across 4 Solution Centers
✓ 40K+ Facilities, 150K Computers, 13K Smartphones, 310K Scanners, 45K POS Terminals, 33PB Storage Capacity
✓ Connects 38K Post Offices and 65K Retail Terminals
✓ 29K Web Pages on .com; 384M Visits Per Year; $840M Online Revenue
✓ 310M Scans/Day; 14K Virtual Servers; 2K Physical Servers
Challenges With IT Projects

- Inconsistent Communication Between IT and the Customer
- Time Spent Developing Large Requirements that Change
- Customers Don’t Always Know Exactly What They Want
- Value of Business Requirements Not Well Defined / Prioritized
- Change is Viewed as Not Being Successful
- Limited Real-Time Transparency on Project Cost and Schedule
- Project Delivery / Quality Needs Improvement
USPS Agile Objectives

**Improve Communication**
- Constant Communication and Collaboration Between the Business and Across IT

**Provide Full Visibility**
- Projects Managed Based on Continuous Inspections & Useful Metrics

**Increase Project Success**
- Projects Completed on Time & Budget in Line with Customer Needs

**Improve Project Quality**
- Teams Use Continuous Integration Software & Automated Testing

**Speed To Market**
- Projects Centered Around Business Value Realized Quickly
Agile Program Scope

- Customer Satisfaction
- Speed to Market
- System Quality
- Project Success

- Business Planning
  - Scrum Methodology
  - Engineering Best Practices

Agile
Business Requirements
- Story Points Assigned
- Prioritized
- Assigned to Releases/Sprints

Prioritized by Product Owners

Daily Tasks managed by the team

Design | Code | Integrate | Test

24 HOURS

2 WEEKS

Creates Sprint Backlog (Technical Requirements)

Demo, Business Partner Signoff, and Retrospective

SIT / CAT Moved to Production
Agile Execution of Lean Six Sigma Changes

Commonalities

• Eliminate waste/rework
• Continuous improvement
• Focus on delivering/improving value for Business Partners

Agile Execution

• Minimizes risk through iterative development and incremental delivery
• Ability to handle change beyond initial process analysis
• Focus and refinement of recommended improvements at the implementation level
• Ideal platform for innovation and new product introduction
• Close coordination between business and IT
USPS Agile Roadmap

- Assess The Organization
- Define Agile Roadmap
- Start Communications
- Change Policy / Agile Training
- Heavy Coaching / Start Engineering
- Controlled Agile Project Selection
- Mandate All System Projects Agile
- Medium Coaching / Metrics / COP
- Expand Agile Planning / Engineering
- Start Agile Business Planning
- PMs Manage Agile Projects
- Light Coaching / Assessments
- End-to-End Automated Testing
- Enterprise SAFe Planning
- Continuous Improvement
USPS Enterprise Agile Status

- IT Policies and Procedures Reviewed and Updated for Agile
- 485 Agile Releases Across 224 Applications
- Standard Agile SW Support with 1685 Users (VersionOne)
- Metrics Developed to Monitor Agile Maturity / Engineering Progress
- Projects Using Engineering Practices (i.e. CI, TDD, Code Quality)
- USPS Continues to Build a Large Agile Knowledge Base
- 1500+ People Trained in Agile (Scrum, Engineering and SAFe)
Two Methodologies

Waterfall Development Methodology Phases

(Plan Driven)

Agile Scrum Development Methodology Phases

(Value Driven)
Project Transparency

Real Time Program Progress / Scope

Scope / Progress Over Time

MS Project Plan

Team Velocity by Sprint

Another View

What If?

Real Time Program Progress / Scope

Status of the Release

Drill

Status of the Sprint

Drill

Status of User Story

Another View

What If?
Agile Engineering Practices

Trained 400+ Development Staff in **Continuous Integration, Test Driven Development**, and **Code Quality Best Practices**

**Training Approach:**
- 18 hours of classroom training / Hands-on Coaching for 4-8 Weeks
- Supplemental Training based on Skill / SW
- Measures Monitored Weekly and Discussed

---

**Open Source Software Tools Being Used**

- **JUnit**: Automated Java Unit Tests
- **Mockito**: Testing Modules in Isolation
- **EclEmma**: Measuring Code Coverage
- **Cobertura**: Measuring Code Coverage
- **DBUnit**: Testing Data Access Layer
- **PLUnit**: Testing PL/sql Code
- **FindBugs**: Static Analysis of Java Violations
- **PMD**: Static Analysis of Java Violations
- **CheckStyle**: Static Analysis of Java Style Violations
- **Jenkins & Sonar**: CI / Dashboard Platform
- **HttpUnit**: Functional Testing of Web Appl
- **Selenium**: Functional Testing of Web Appl / Browser
- **Ant**: Build Scripting Tool
- **Maven**: Dependency Mgmt and Build Tool
- **Gradle**: Dependency Mgmt and Build Tool

---

**Training Courses Provided**

- Unit Testing with JUnit 4
- Test-Driven Development with JUnit
- Testing Classes in Isolation with Mockito
- Functional Web Testing with HttpUnit
- Functional Web Testing with Selenium WebDriver
- Improving the Structure of a System with Refactoring
- Improving Code Quality with SOLID design principles
- Automated Build Principles and Practices
- Continuous Integration with Jenkins
- Continuous Integration with Ant and Gradle
# Engineering Levels

<table>
<thead>
<tr>
<th>Training</th>
<th>Novice</th>
<th>Journeyman</th>
<th>Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Receives Overview of USPS Development Standards</td>
<td>• Practices USPS Development Standards</td>
<td>• Understands Simple Design</td>
<td>• Practices No Big Design Up Front</td>
</tr>
<tr>
<td>• Learns Basic TDD</td>
<td>• Practices Collaborative Code Ownership</td>
<td>• Participates in Peer Code Reviews</td>
<td>• Practices Pair Programming</td>
</tr>
<tr>
<td>• Learns about Team Development</td>
<td>• Practices TDD</td>
<td>• Practices Regular Refactoring</td>
<td>• Uses Build Reports Daily</td>
</tr>
<tr>
<td>Test</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Learns JUnit 4</td>
<td>• Uses Automated Unit Tests</td>
<td>• Uses Automated Functional Tests</td>
<td>• Integrates Test Output with Bug Tracking System</td>
</tr>
<tr>
<td>• Learns how to create Mock Objects with Mockido</td>
<td>• Always Tests Before Committing</td>
<td>• Uses Static Code Analysis</td>
<td>• Increases Tests based on Risk</td>
</tr>
<tr>
<td>• Understands Different Types of Tests</td>
<td>• Automates Nightly Snapshot Builds</td>
<td>• Improves Code Coverage</td>
<td>• Automates Regression Tests</td>
</tr>
<tr>
<td>• Learns Test Automation</td>
<td>• Automates nightly snapshot builds</td>
<td>• Leverages Jenkins Heavily</td>
<td>• Automates Build Output Reports</td>
</tr>
<tr>
<td>Build</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Learns what Continuous Integration is</td>
<td>• Uses Automated Builds</td>
<td>• Automates Dependency Management</td>
<td>• Integrates Additional Reporting and Feedback Mechanisms</td>
</tr>
<tr>
<td>• Learns how to leverage tools for Continuous Integration</td>
<td>• Outputs Build Process Reports</td>
<td>• Automates Build Output Reports</td>
<td></td>
</tr>
<tr>
<td>Deploy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Learns how to Automate Builds</td>
<td>• Uses Helper Scripts for Deployments</td>
<td>• Uses Automated Scripts</td>
<td>• Leverages Jenkins for Automated Deployments</td>
</tr>
<tr>
<td>Visibility</td>
<td>• Uses Knowledge Sharing Tools</td>
<td>• Uses Automated Deployments (DEV)</td>
<td>• Produces Unified Output included release notes and bug reports</td>
</tr>
<tr>
<td>• Learns how to use Tool Output</td>
<td>• Leverages Tool Based Reports</td>
<td>• Uses Manual Deployments (SIT, CAT, PROD)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Automates Build Process Output</td>
<td>• Uses Reports as Input</td>
<td>• Project Outputs are Visible to entire Organization</td>
</tr>
<tr>
<td></td>
<td>• Uses Reports as Input</td>
<td></td>
<td>• Project Outputs are Used as Team Measuring</td>
</tr>
</tbody>
</table>
SAFe is a proven, publicly available framework for applying Lean / Agile practices at the **enterprise scale**

- **SAFe** is intended for large teams of 50+ people and there are significant business and/or technological dependencies or coordination.
- It's a framework that aims to synchronize vision, planning, interdependencies, and alignment of business across an organization and delivery teams.
- USPS recently implemented **SAFe** on one of its largest revenue system suites involving over 300 staff.
Mandating Agile Practices May be Necessary

Executive Management Must Support and Encourage

Assist the Customer in Embracing Agile

Use Metrics to Monitor Progress and Maturity

Mandating Agile Practices May be Necessary

Experienced Coaching is Critical – Don’t Just Train

Higher Risk of Failure Blending Waterfall & Agile

Do Not Modify Agile Processes Around Existing SW

Communicate Program Progress, Direction, & Successes
Realized Benefits

- More Projects Completing On-Time / Within Budget
- Issues Identified Earlier and Being Escalated
- Improved Project Communications Overall
- Customer Addressing High Business Value First
- Starting to See Code Quality Improvement
- Increased Customer Satisfaction
- Increased Team Satisfaction and Productivity
Challenges Continue

- Continued Resistance to Change
- Dedicated Project Resources
- Expanding Agile on Large Projects
- Change in Employee Skills / Job Descriptions
- Embedding into Procurements (ex: FFP Contracts)
How Do We Stack Up With Industry?

**Gartner * **

**Modern Application Development**

- Agile (Scrum/Kanban)
- **Modern Tools**
  - Scrum Boards
  - Static Analysis
  - ADLM Tools
- **Agile Project Management at the enterprise level**
  - SAFe
- **DevOps Tools**
  - Continuous Integration
  - Automated Testing
  - Automated Deployment
  - Private IaaS/PaaS
  - Monitoring


**USPS Development Environment**

- Agile (Scrum/Scrumban)
- **Modern Tools**
  - Scrum Boards
  - Static Analysis (Sonar)
  - ADLM Tools (HP/ALM & VersionOne)
- **Agile Project Management at the enterprise level**
  - SAFe
- **DevOps Tools**
  - Continuous Integration (Jenkins)
  - Automated Testing (HP/QTP/PC/Selenium)
  - Automated Deployment (Future)
  - Private IaaS/PaaS (VMware)
  - Monitoring (AppDynamics)

* Source: Gartner – Modern Application Development Life Cycle Management.*
What’s Next?

- Expand Agile / Engineering to All Projects
- Measure, Baseline, And Continuously Improve
- Continue to Train and Coach IT Employees in Agile
- Expand Agile at the Business Planning Level Using SAFe
Questions

mark.a.mittelman@usps.gov