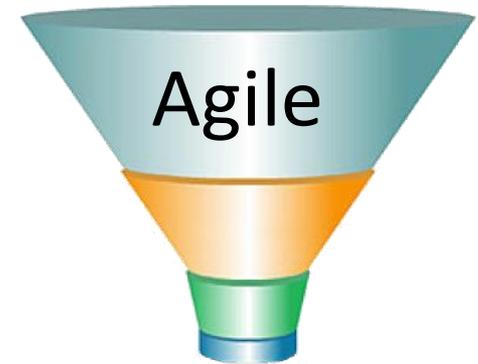


AGILE DEVELOPMENT

U.S. POSTAL SERVICE

FEBRUARY 19, 2015



Business Solutions



Agenda

● State of Software Projects

● USPS Overview

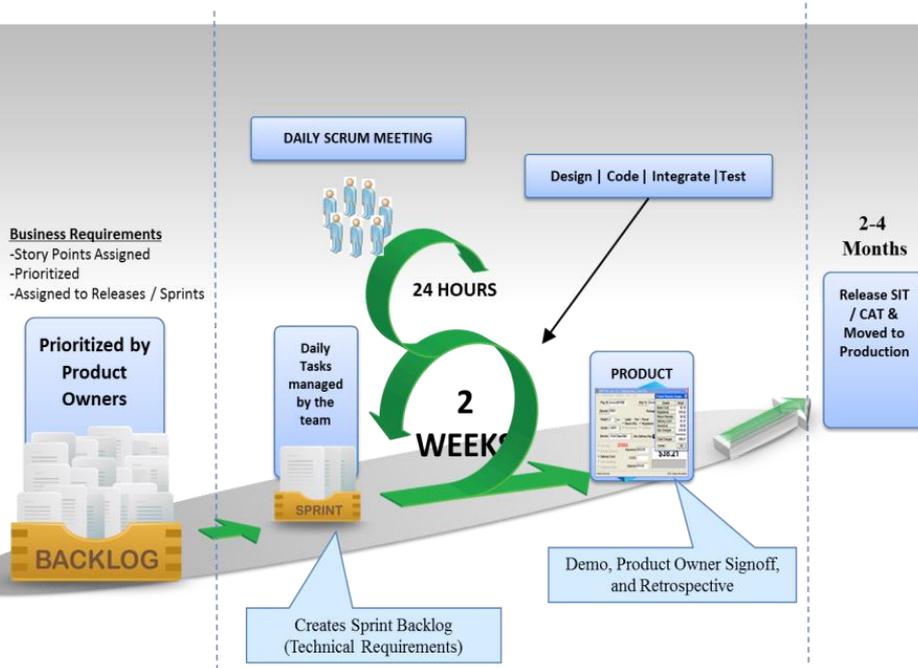
● Agile Program

● Agile Engineering

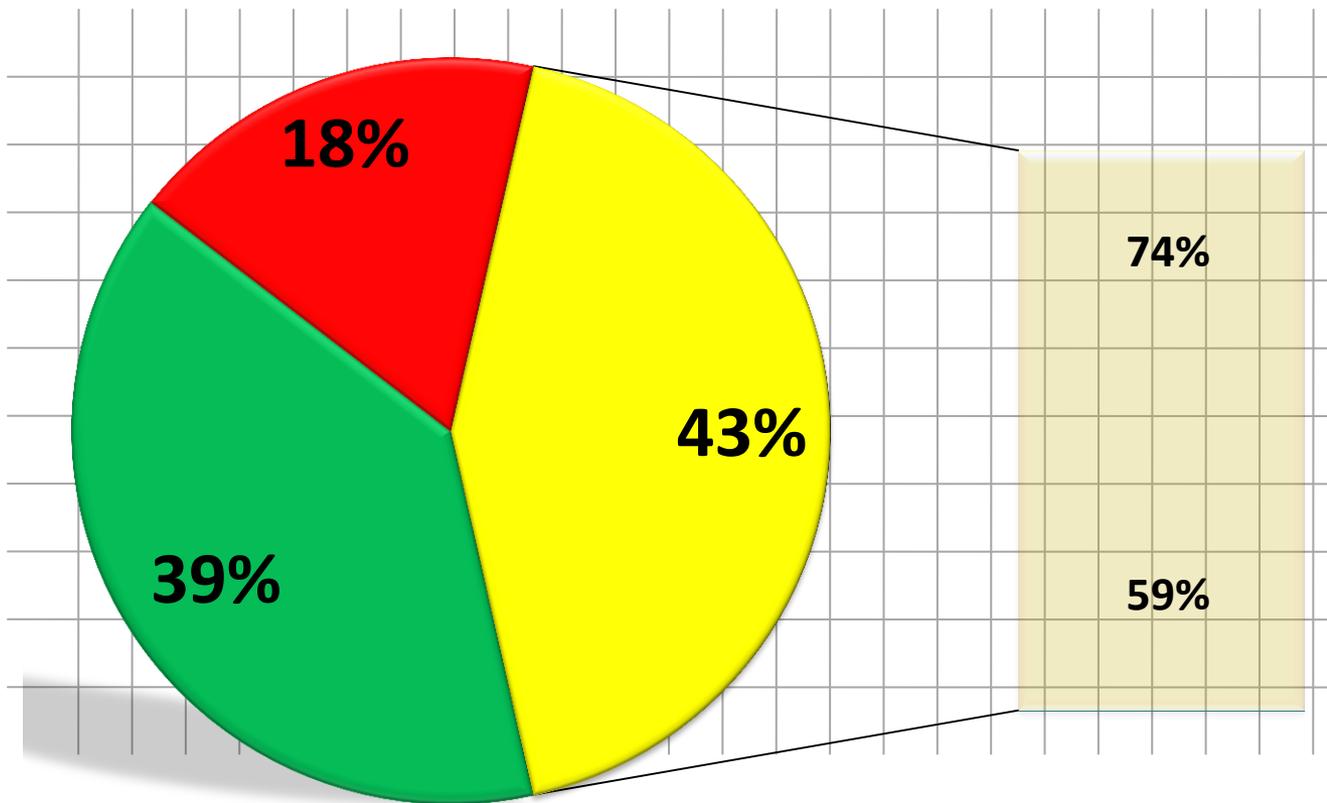
● Scaled Agile Framework

● USPS Retrospective

● What's Next for the USPS



Industry Situation



What drove us here – industry-wide view

■ 43%

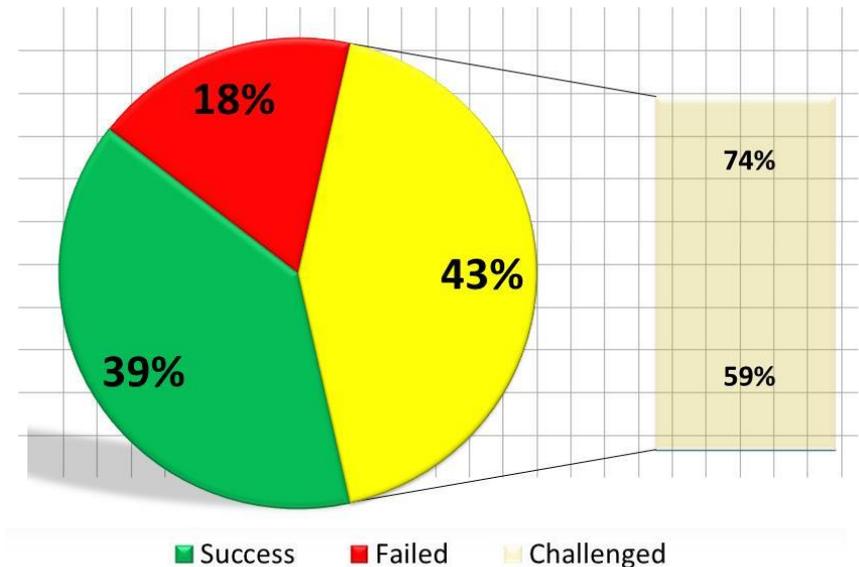
- Challenged: Late, Over Budget, or Missing Key Features
- 74% - schedule overruns
- 59% - cost overruns

■ 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

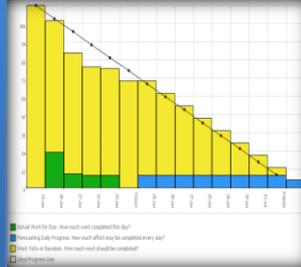
USPS Agile Objectives

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

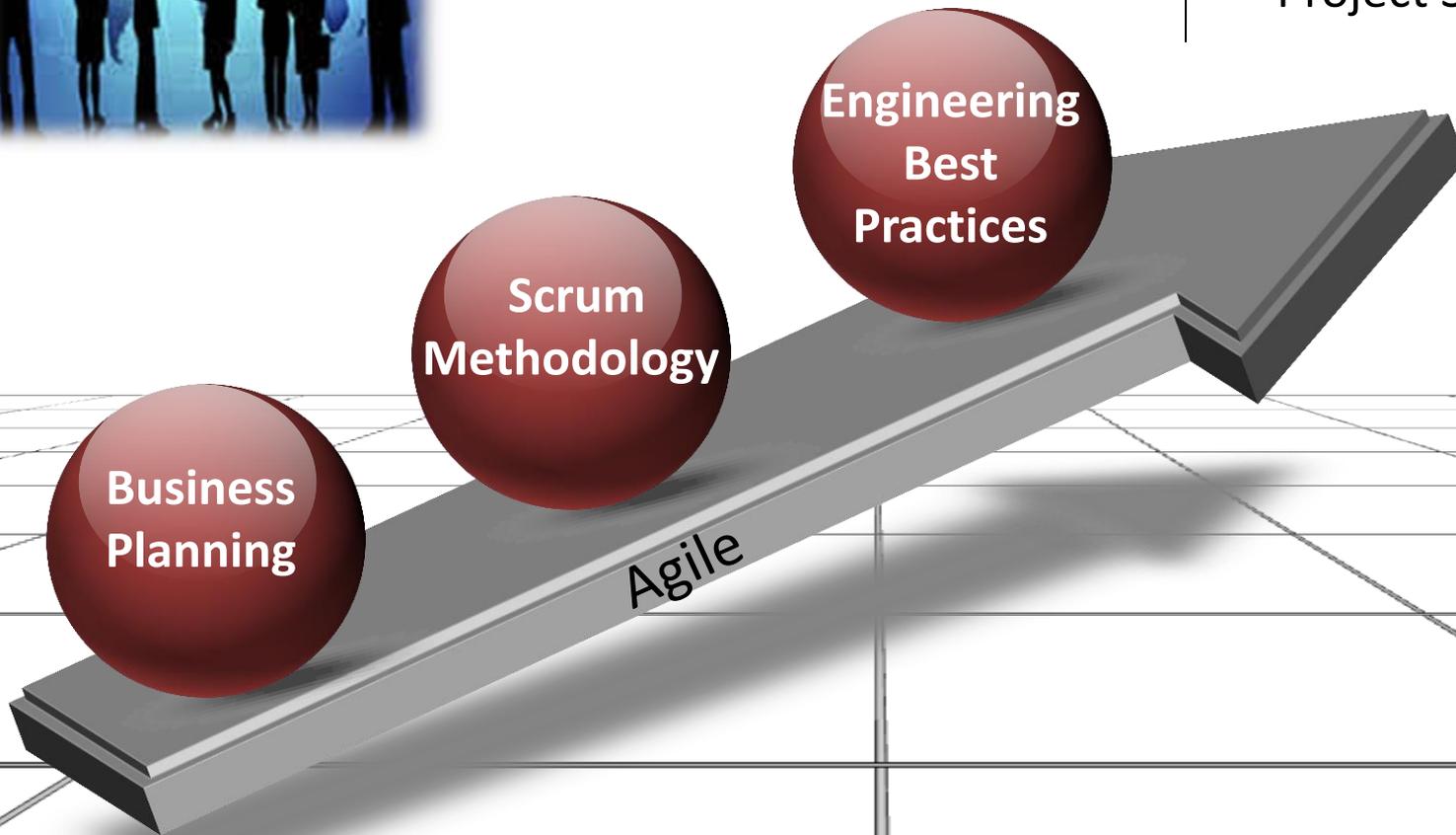


Projects Centered Around Business Value Realized Quickly

Agile Program Scope



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



Agile Process Flow

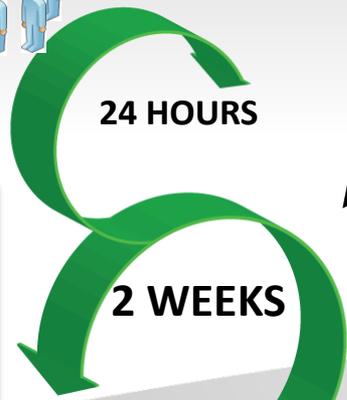
Business Requirements

- Story Points Assigned
- Prioritized
- Assigned to Releases/Sprints

DAILY SCRUM MEETING



Design | Code | Integrate | Test



Prioritized by
Product
Owners



Daily
Tasks
managed
by the
team

SPRINT

Creates Sprint Backlog
(Technical Requirements)



Demo, Business Partner
Signoff, and Retrospective

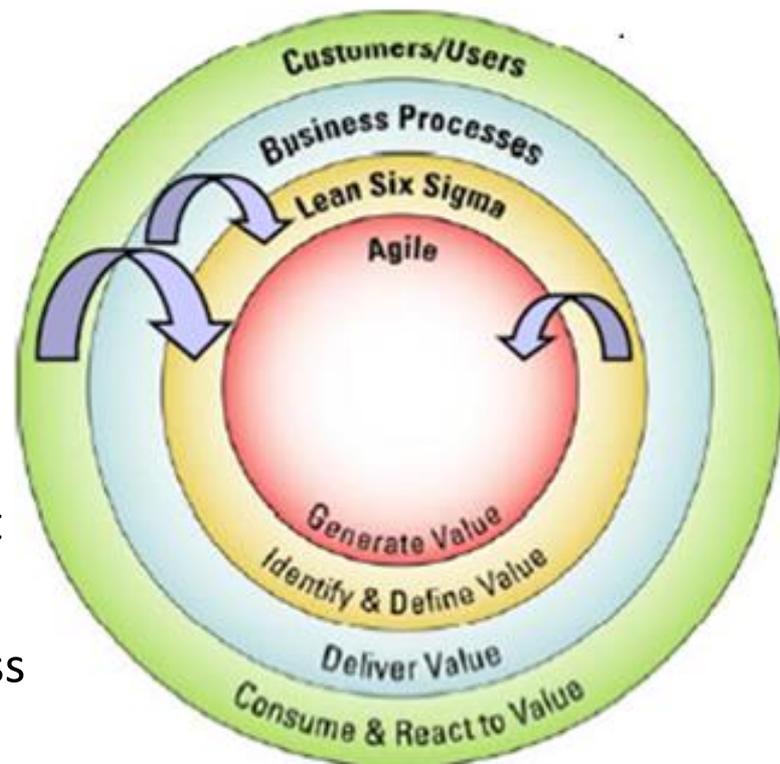
SIT / CAT
Moved to
Production

Commonalities

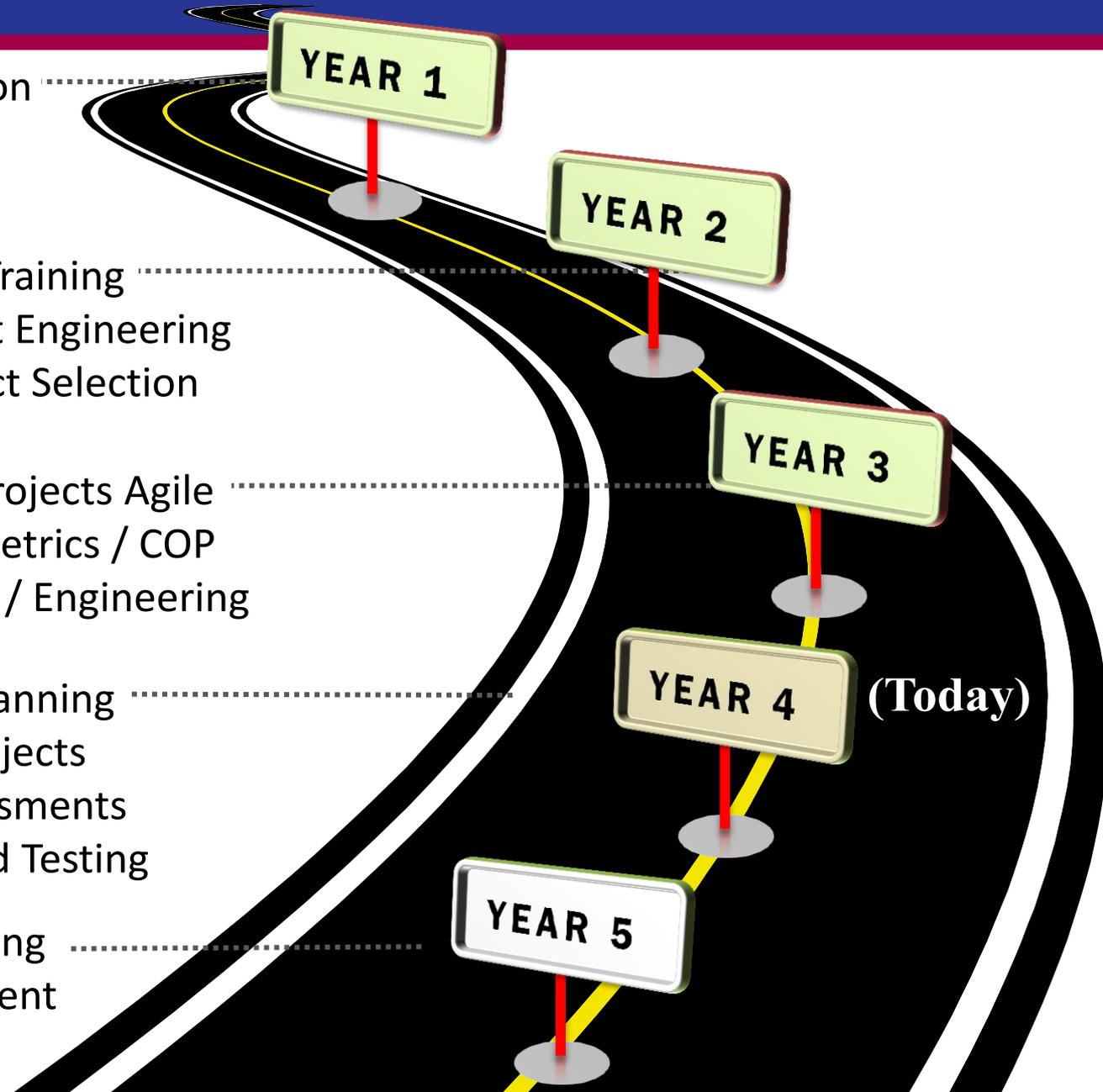
- Eliminate waste/rework
- Continuous improvement
- Focus on delivering/improving value for Business Partners

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



- 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

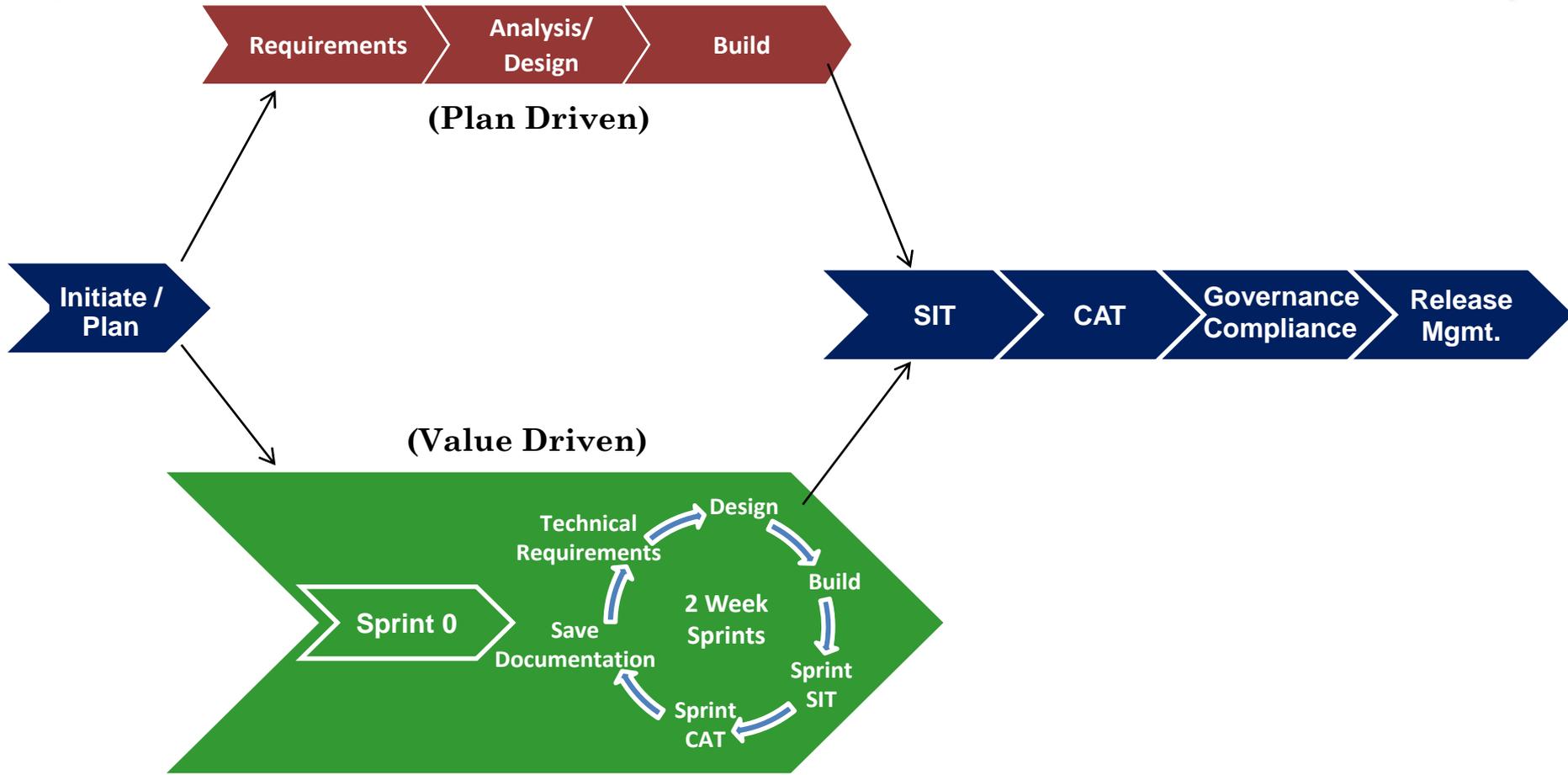
Projects Using Engineering Practices (i.e. CI, TDD, Code Quality)

USPS Continues to Build a Large Agile Knowledge Base

1500+ People Trained in Agile (Scrum, Engineering and SAFe)

Two Methodologies

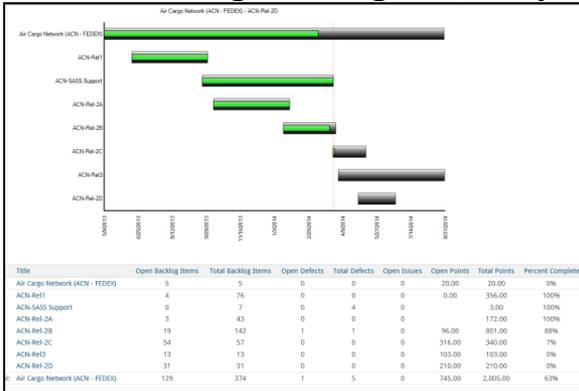
Waterfall Development Methodology Phases



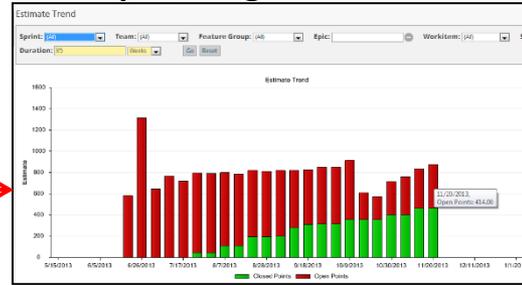
Agile Scrum Development Methodology Phases

Project Transparency

Real Time Program Progress / Scope



Scope / Progress Over Time

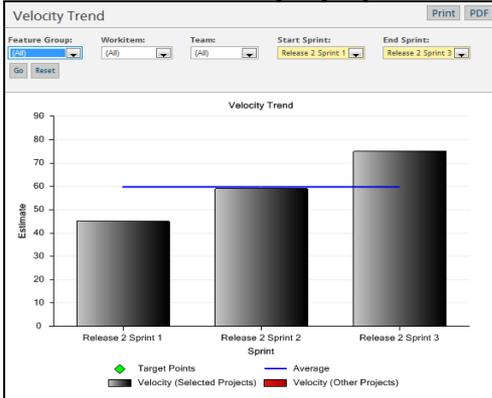


Another View →

MS Project Plan

WBS	Task Name	Start	End	Progress
1	Sample Agile Project Plan Template			
1.1	Initiate and Plan			
1.2	Sprint 0 - Analysis and Design			
1.3	Development - Sprint 1 through n			
1.4	Systems Integration Test			
1.5	Customer Acceptance Test			
1.6	Governance			
1.7	Release			

Team Velocity by Sprint



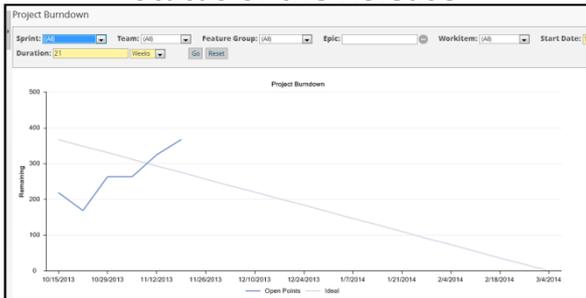
What If? →

Real Time Program Progress / Scope



WBS	Task Name	Start	End	Progress
1	Initiate and Plan			
1.1	Business Needs Statement - Draft/Review/Initiate DNS			
1.2	Develop Cost and Schedule Rough Order of Magnitude (ROM)			
1.2.1	Draft/Review/Initiate ROM Estimate			
1.2.2	Approve / Cancel Project in Team Track			
1.2.3	Draft/Review/Initiate Project Plan to Define Project Requirements			
1.2.4	Draft/Review/Initiate Cost Matrix - Requirements & Design			
1.3	Obtain ROM and Schedule Approvals for Funding			
1.3.1	Obtain IT Approval for ROM and Requirements and Design Plan			
1.3.2	Requirements and Design ROM and Schedule - Customer Approval			
1.3.3	Obtain Funding Authorization Information			
1.3.4	Obtain "Approval for Requirements and Design" Authorization/Signature			
1.4	TMOS Project Plan			
1.4.1	Create Draft MS Project Plan for TMOS			
1.4.2	Review TMOS Project Plan / Data with all Project Stakeholders			
1.4.3	Update TMOS System with Project Plan and Project Data			
2	Sprint 0			
2.1	Obtain ROM Funding			
2.2	Create Requirements / Obtain Approval			
2.2.1	Create / Finalize Requirements (Product Backlog - User Stories, Story Points, Prioritization, etc)			
2.2.2	Receive Requirements Approval in Writing (Letter)			
2.3	Business Impact Assessment - BIA			
2.3.1	Draft/Review BIA (using eCA)			
2.3.2	IT Solution Stakeholders Review			
2.4	Project Notification - EAC Checklist 1			
2.5	Develop Draft Approved Architecture			
2.5.1	Develop Draft Architecture			
2.5.2	Obtain IT Approval for Draft Architecture			

Status of the Release



Status of the Sprint



Status of User Story



Drill →

Drill →

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

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

Open Source Software Tools Being Used

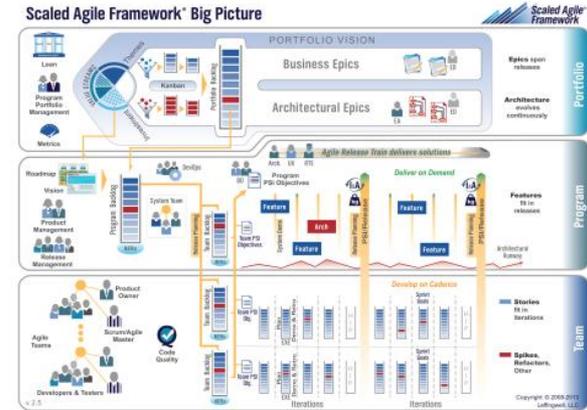
JUnit	Automated Java Unit Tests
Mockito	Testing Modules in Isolation
EclEmma	Measuring Code Coverage
Cobertura	Measuring Code Coverage
DBUnit	Testing Data Access Layer
PLUnit	Testing PL/SQL Code
FindBugs	Static Analysis of Java Violations
PMD	Static Analysis of Java Violations
CheckStyle	Static Analysis 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

Engineering Levels

	Training	Novice	Journeyman	Expert
Develop	<ul style="list-style-type: none"> Receives Overview of USPS Development Standards Learns Basic TDD Learns about Team Development 	<ul style="list-style-type: none"> Practices USPS Development Standards Practices Collaborative Code Ownership Practices TDD 	<ul style="list-style-type: none"> Understands Simple Design Participates in Peer Code Reviews Practices Regular Refactoring 	<ul style="list-style-type: none"> Practices No Big Design Up Front Practices Pair Programming Uses Build Reports Daily
Test	<ul style="list-style-type: none"> Learns JUnit 4 Learns how to create Mock Objects with Mockito Understands Different Types of Tests Learns Test Automation 	<ul style="list-style-type: none"> Uses Automated Unit Tests Always Tests Before Committing 	<ul style="list-style-type: none"> Uses Automated Functional Tests Uses Static Code Analysis Improves Code Coverage 	<ul style="list-style-type: none"> Integrates Test Output with Bug Tracking System Increases Tests based on Risk Automates Regression Tests
Build	<ul style="list-style-type: none"> Learns what Continuous Integration is Learns how to leverage tools for Continuous Integration 	<ul style="list-style-type: none"> Uses Automated Builds Outputs Build Process Reports 	<ul style="list-style-type: none"> Automates Nightly Snapshot Builds Leverages Jenkins Heavily Automates Dependency Management 	<ul style="list-style-type: none"> Automates Build Output Reports Integrates Additional Reporting and Feedback Mechanisms
Deploy	<ul style="list-style-type: none"> Learns how to Automate Builds 	<ul style="list-style-type: none"> Uses Helper Scripts for Deployments 	<ul style="list-style-type: none"> Uses Automated Scripts Uses Automated Deployments (DEV) Uses Manual Deployments (SIT, CAT, PROD) 	<ul style="list-style-type: none"> Leverages Jenkins for Automated Deployments Produces Unified Output included release notes and bug reports
Visibility	<ul style="list-style-type: none"> Learns how to use Tool Output 	<ul style="list-style-type: none"> Uses Knowledge Sharing Tools Leverages Tool Based Reports 	<ul style="list-style-type: none"> Automates Build Process Output Uses Reports as Input 	<ul style="list-style-type: none"> Project Outputs are Visible to entire Organization Project Outputs are Used as Team Measuring

Scaled Agile Framework (SAFe)

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
- Its 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

Lessons Learned



Does Not Solve Org. Impediments – It Exposes Them

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

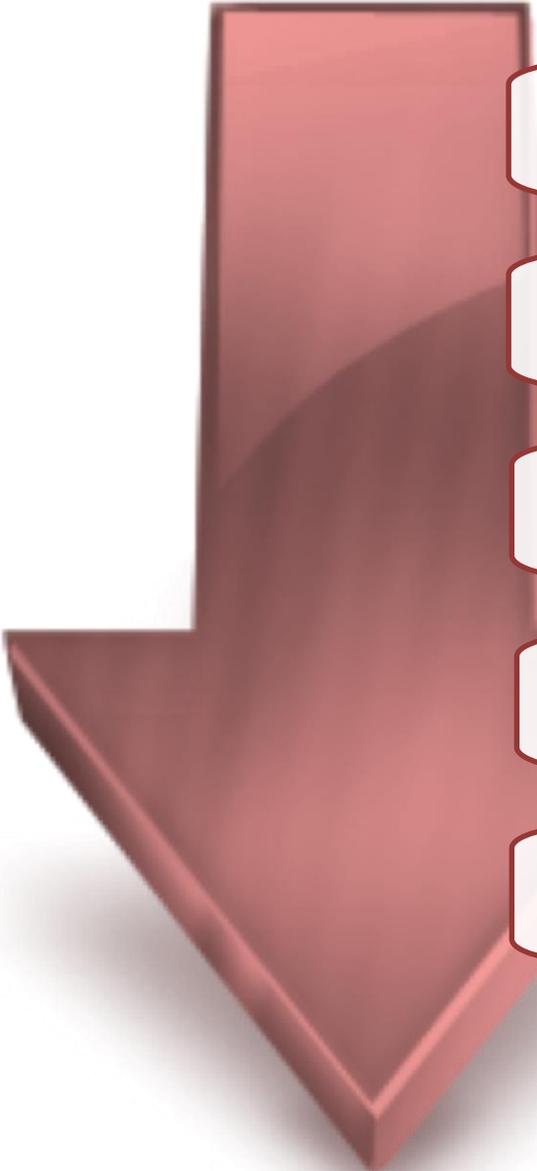
Do Not Modify Agile Processes Around Existing SW

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)

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)

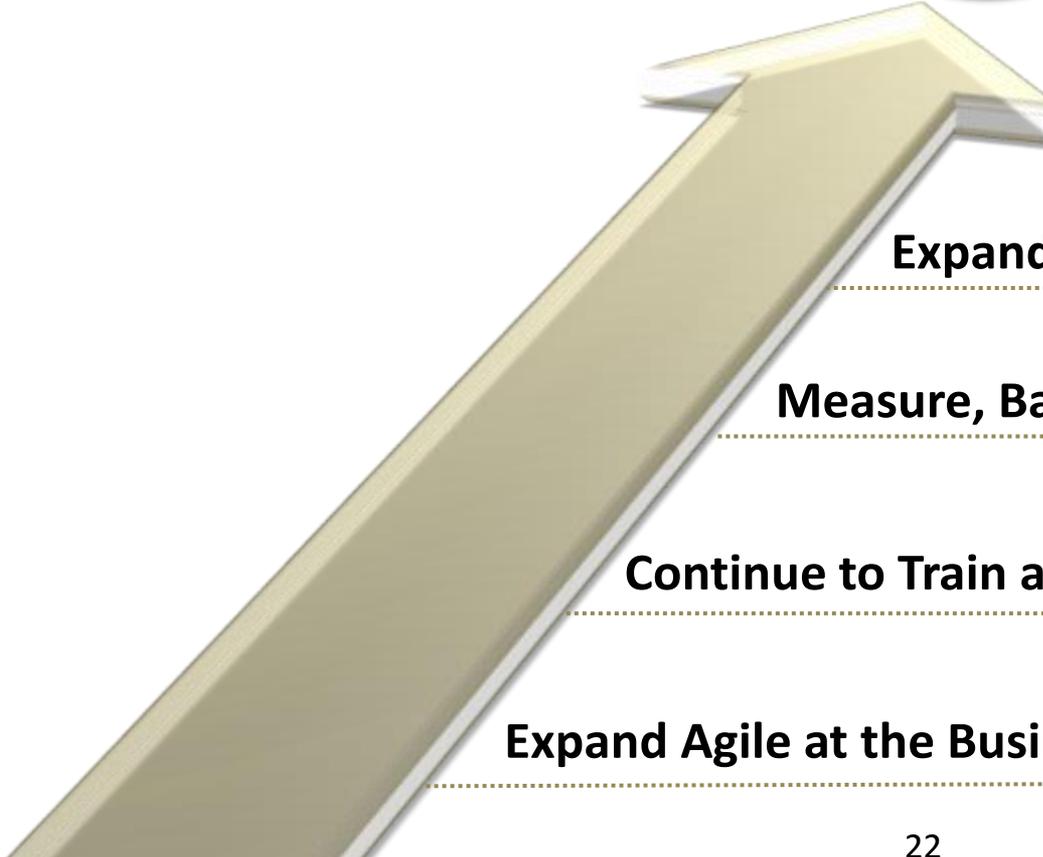
* Source: Gartner – Modern Application Development Life Cycle Management.

What's Next?

**Speed To
Market**



**Quality
Business
Solutions**



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



UNITED STATES
POSTAL SERVICE™