

ATK, Grassroots implementation of PSP

Dr. Jason Brady

Director of Business Systems,
Alliant TechSystems, Inc. (ATK)



The Customers and Markets We Serve



ATK is a Fortune 500 aerospace, defense, and commercial products company with operations in 21 states, the Dominican Republic, Puerto Rico, and internationally



**Soldier
Systems**



**Rotary-Wing
Military Aircraft**



**Fixed-Wing
Military Aircraft**



**Commercial
Aerospace**



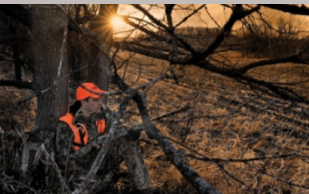
Satellites



**Human Space
Launch**



**Special
Operations
Forces**



Sport Shooting



**Law
Enforcement**



**Ground Combat
Vehicles**



**Naval
Platforms**



**Satellite
and Strategic
Launch**

- World's top producer of solid rocket propulsion systems
- World's largest producer of military ammunition
- Leader in affordable precision weapons, propellants, and energetics
- Leading brands in law enforcement and sporting ammunition
- Leading brands in soldier systems, sporting, and hunting accessories
- Provider of advanced composite structures, satellite components, and subsystems
- Expertise in managing and operating Government-owned facilities

Aerospace Group



Defense Group



Sporting Group



ATK Offices and Operating Locations



Alabama

Huntsville

Arizona

Mesa

California

Commerce

Goleta

Laguna Hills

Monterey

Oroville

Pasadena

Rancho Bernardo

San Diego

Vandenberg AFB

Woodland Hills

Florida

Cape Canaveral

Clearwater

Kennedy Space Center

Palm Beach Gardens

Idaho

Lewiston

Meridian

Indiana

Richmond

Maryland

Aberdeen Proving Ground

Baltimore

Beltsville

Elkton

Massachusetts

Hopkinton

Minnesota

Anoka

Eden Prairie

Elk River

Plymouth

Mississippi

Iuka

Missouri

Fenton

Independence

Montana

Bozeman

New Jersey

Picatinny Arsenal

New Mexico

Socorro

New York

Ronkonkoma

North Carolina

Fayetteville

Southport

Ohio

Dayton

Texas

Fort Worth

Houston

Corporate headquarters and offices

Arlington, VA

Eden Prairie, MN

Sporting Group headquarters

Anoka, MN

Defense Group headquarters

Baltimore, MD

Aerospace Group headquarters

Magna, UT

Puerto Rico



Utah

Brigham City

Clearfield

Logan

Magna

West Virginia

Rocket Center

Dominican Republic

Santo Domingo

Puerto Rico

Lares

Mayaguez

A long time ago... we had a significant amount of issues within Software Development and Enterprise Systems:

- Deadlines on tasks were often missed and almost all business systems projects were delivered late.
- Staff would consistently complain about priority changes, un-reasonable deadlines, management imposed deadlines.
- Enormous time was spent on production support.
- Issues were not being escalated – and if they were, weren't being resolved.
- New tasks and projects were assigned and resources reallocated prior to work being completed.
- Work was often disbanded and never resumed – no effective value provided to the business.
- Integrations, custom applications, could not be re-used across organizations due to divergent process requirements.

What was tried...

- Project request forms were created to get key information.
- Task and project lists were created.
- Requirements and test plans had a hard signoff to drive accountability.
- Implemented gated project management solution.
- Standardized approach and started to “reuse” code and share code across organizations.

What we got...

- Forms that weren’t filled out, with users complaining.
- Just one more list of activities that wasn’t up to date.
- Oops, I’m sorry, missed that requirement. I still need it for the business.
- A schedule to update, that was always behind. Not on-time, let’s just change the dates.
- Poor quality code was just used across the organizations.

Managers and employees were often frustrated. Heroics and conflict became common place.

We weren't getting anything done...



- We were often told, just allocate 50% of your time to the project.
- Multiple projects would be split up, everything was late, nothing completed.
- Managers would just decide to “move on” and not finish the prior activity.

Objectives for Week:	Expected Hours	Actual Hours	Variance
TMS	5	3.25	-1.75
DistCode	5	3	-2
		6.25	

Date	Time on T	Task Type	Project Name	Project Task	Scheduled
9/26/2011	60	Research	TELIST Accessment		Yes
9/26/2011	60	Documentation	TMS	Deployment Release	Yes
9/26/2011	90	Production Release	Store Scheduler	Production Release	Yes
9/26/2011	60	Meeting	Meeging		Yes
9/26/2011	30	Issue	PayRoll	PayRoll Issue	No
9/26/2011	30	Issue	TMS	Rick McHugh	No
9/26/2011	30	Research	TELIST Accessment		No
9/26/2011	30	Support	TMS		No
9/26/2011	60	Support	ESB	Meeting Attendance	No
9/26/2011	15	Support	PSDB	Support - Tillman	No
9/26/2011	30	Support	PayRoll	Payroll Processing	No

Seriously, you said that? A maximum of 15hrs time on task a week?

Actions:

- Employees tracked time on task and time off task for two weeks.
- Sought to improve TOT during second week.
- Determined available ‘value added’ work for the organization based on nominal workload.

9/27/2011	30	Meeting	Kaizen		Yes
9/27/2011	45	Disruption	WorkStation Reboot		No
9/27/2011	15	Support	IDAutomation - Barcode	BT	No
9/27/2011	60	Documentation	TMS	Deployment Release	Yes
9/27/2011	15	Meeting	ESB	Admin Call	Yes
9/27/2011	30	Research	TELIST Accessment		Yes

Average TOT for individual was 5-10hrs for week 1, 10-15 hrs for week 2.

9/27/2011	30	Meeting	Distributed Code		No
9/27/2011	30	Meeting	Tag Up		No
9/28/2011	15	Support	Account	BS	No
9/28/2011	15	Support	Shipping	RM	No
9/28/2011	20	Meeting	ESB		No
9/28/2011	120	Research	TELIST Accessment		Yes

Determine SLA and Address Capacity



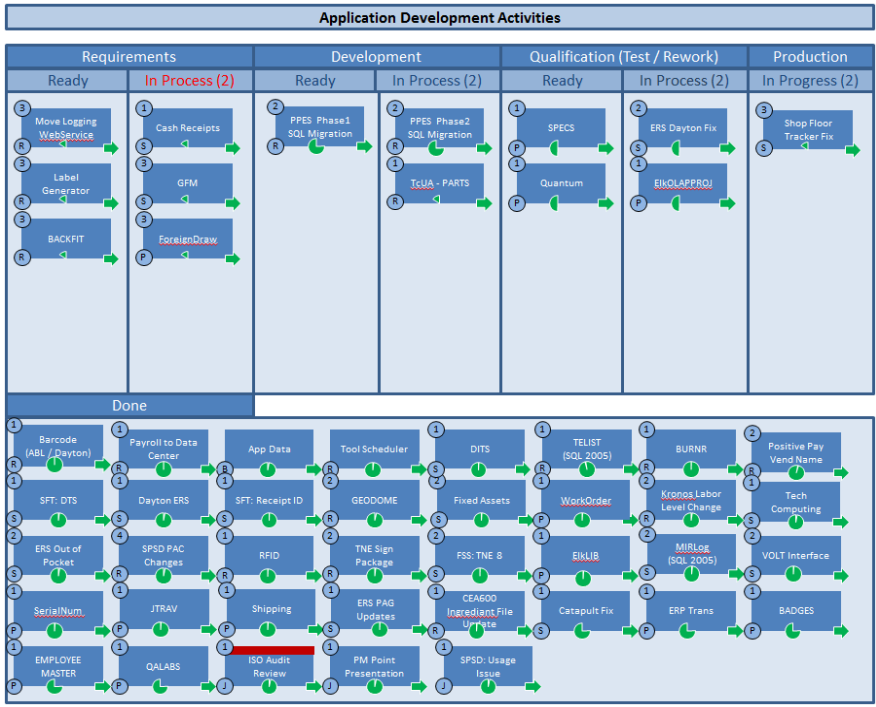
The first major improvement was to determine service expectations, identify capacity and limit WIP.

Actions

- Identified service level agreements for each work type.
- Identify services provided to organization.
- Define “acceptable” delivery time.
- Level set delivery time against Time on Task.

Current State:

- Projects are now pulled into each phase of software development.
- Natural breaks for priority changes
- Resources are focused on finite deliverables.
- Visual indication of “stopped work” with immediate escalation.
- Resources work issues rather than project shift.
- More predictable schedules established.



	Initial State	Current State
Active Projects in Team.	15+	6
Average Stop Work Issue Resolution	5-10 Days	1-2 Days
Cycle Time	10-12 Weeks	3 Weeks
Time on Task	9hrs p/week	12hrs p/week

We needed to improve, so we baseline...



Any deviation is a defect?

Projects were baselined. Learning from changes, defects, and issues, we can improve our development lifecycle to improve quality, reduce development costs, and meet expectations.

Use the Project Change Requests list to facilitate requests for significant changes to the project scope.

Type	Q	Title	Change Requested By	Change Request Date	Date Due	Issue Indicator	Days Impact To Project	Change Request Status	Defect	Defect Stage
		Training tracker button	Hagelin, Jill							
		Change Date name	Hagelin, Jill							
		Too Many Attachments	Hagelin, Jill							
		Running a report	Hagelin, Jill							
		System Index error NEW	Hagelin, Jill							
		Deleting an employee NEW	Hagelin, Jill							
		Combining Tabs	Perrine, Favin							
		Highlight which tab is in use.	Perrine, Favin							
		Addition of FY entry.	Perrine, Favin							
		Project Definition	Perrine, Favin							
		Cost Field	Perrine, Favin							
		Identification of Internal vs External	Perrine, Favin							
		Class Hours Specification	Perrine, Favin							
		Export on Search Tab	Perrine, Favin							
		Account Code Field	Perrine, Favin							
		Record Training Tab - Opening View	Perrine, Favin							
		Server error message	Hagelin, Jill							

Project Planning	Improve Requirement	Elicitation	Prototypin	Improve Test Procedure	Developer Training
			Yes		
			Yes		
				Yes	
	Yes	Yes			
	Yes		Yes		
	Yes		Yes		
	Yes		Yes		
	Yes	Yes			
	Yes	Yes			
	Yes	Yes			
					Yes
	Yes		Yes		
	Yes		Yes		
			Yes		
				Yes	
	Yes		Yes	Yes	
	Yes		Yes	Yes	

All defects, issues, changes are logged during project.

Types of issues are quantified.

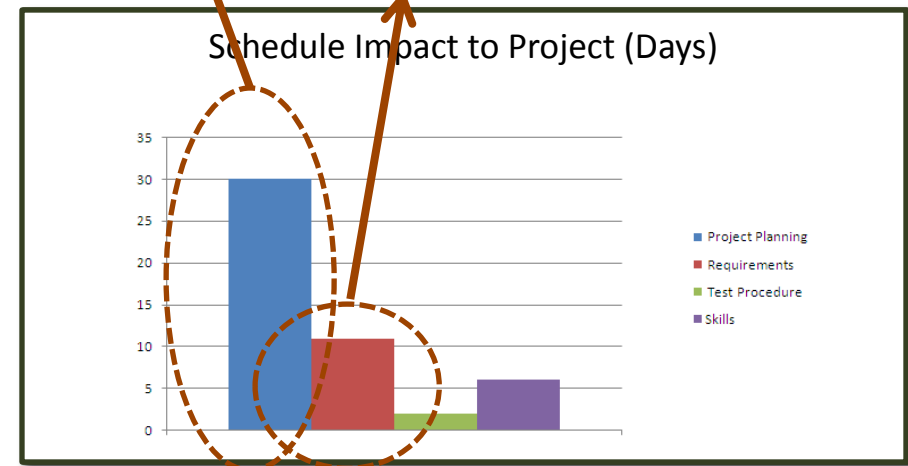
Target areas identified for improvement.

Action plan established to improve.

Time on Task started to increase.

Improve risk management

Improve requirement definition to incorporate prototyping



Needed to strengthen weak areas in process.



Kanban board made work visible. We would overlay, week after week on top of each other. Then, trends could be seen...

- We were now getting stuck on areas that weren't part of our "core competencies".
- Employees avoid these areas, as they don't know what to do, or a high amount of issues arise.
- Scripts were established to guide individual on how to "best address" a given situation. These exist for project kickoff, design, test, production release.
- Kickoff meetings we're added to address problematic areas.

Testing/Quality KickOff Script

Purpose	To guide the Topic Content of the Testing/Quality KickOff Meeting
Entry Criteria	<ul style="list-style-type: none"> o A Completed and Reviewed Problem Statement / Requirements o A Completed Implementation for the Item under Test which has gone through both a Design and Code Reviews. o An Available Test Environment into which the Item can be deployed o A Completed Test Plan



Step	Activities	Description
1	Establish Test Team	<ul style="list-style-type: none"> o Contact Application Owner(s) and request the identification of individuals who will be responsible for performing application qualification tests. Each site will need to provide resources unless otherwise indicated by that site's SME.
2	Deployment	<ul style="list-style-type: none"> o Deploy the Application into the Testing Environment
3	Schedule Meeting	<ul style="list-style-type: none"> o Each site that will use the application will need to be included on the required attendee list. o Each site will require a SME and tester(s). o Provide both a copy of the Requirements Statement and the Test Plan in your meeting request and ask that participants review the content prior to the meeting. o If available, provide a user guide or help pages to assist in allowing the testers to understand how to use the application in question.
4a	Conduct Meeting	<ul style="list-style-type: none"> o Provide an introduction with the purpose of the meeting
		participants why they have been asked to
		f summary of the requirements. Detail is t here. Strive for a general understanding of nd/or functions.
		st plan.

We became effective at setting and managing expectations with other departments and organizations. Even in areas where "it wasn't our job".

To improve quality, we added checklists



Our attention turned to raising the level of quality. We modified the processes to be “learning processes”. This included standards and checklists.

- Personal checklists were added to improve quality of deliverables throughout the process.
- The personal checklist is based on the prior performance of the individual.
- Adding this, was the single most contribution to improve software quality. The average defect yield increased from 40% to 80%.

So, we created checklists for everything!

Code Review Checklist Template

Employee _____ Date _____
Program _____ Revision _____
Project Mgr. _____ Language _____

Purpose	To guide you in conducting an effective code review
General	<ul style="list-style-type: none">- Review the entire program for each checklist category; do not attempt to review for more than one category at a time!- As you complete each review step, check off that item in the box at the right.- Complete the checklist for one program or program unit before reviewing the next.

Defect Log #	Review Category	Review Details		Module 1	Module 2	Module 3	Module 4
	Complete	Verify that the code covers all of the design					
	Calls						
9	DB Objects						

Network Refresh Checklist

Overview:

This document outlines the steps required for verification and installation of the Operational Status of the new Switch.

Revision History

Date	Changed By	Description of Change
5/7/2012		Initial Document
5/29/2012		Updated VTP information

Switch Name: _____ Building #: _____ Date: _____

Item	Task
New Switch Configured	<input type="checkbox"/> Old Configuration saved..... <input type="checkbox"/> Configuration copied to new switch <input type="checkbox"/> Show Run to compare Configuration <input type="checkbox"/> Fiber Modules Installed <input type="checkbox"/> Generate Crypto Keys for SSH <input type="checkbox"/> VTP mode client <input type="checkbox"/> VTP domain ELKTONVTP <input type="checkbox"/> VTP version 3
Switch Labeled	<input type="checkbox"/> IP Address <input type="checkbox"/> Host Name <input type="checkbox"/> Lease Number <input type="checkbox"/> Installation Date
Communication	<input type="checkbox"/> Building Supervisor Contacted <input type="checkbox"/> Supervisor Name _____ <input type="checkbox"/> Supervisor Phone Number _____ <input type="checkbox"/> Data Supervisor Contacted

- Defect yield rate on application development projects is increasing.
- Discussions are occurring on resource allocation on a daily basis, allocation issues are being solved 1-2 weeks in advanced.
- Project milestones often completed ahead of schedule.
- Communication is increasing.
- Emergency work – hair on fire – work is decreasing overall.
- We can talk about defects and performance without everyone being upset.
- Most importantly, pride and satisfaction are increasing.

Our Division has been cited on several audits for best practices in software process and LEAN techniques.

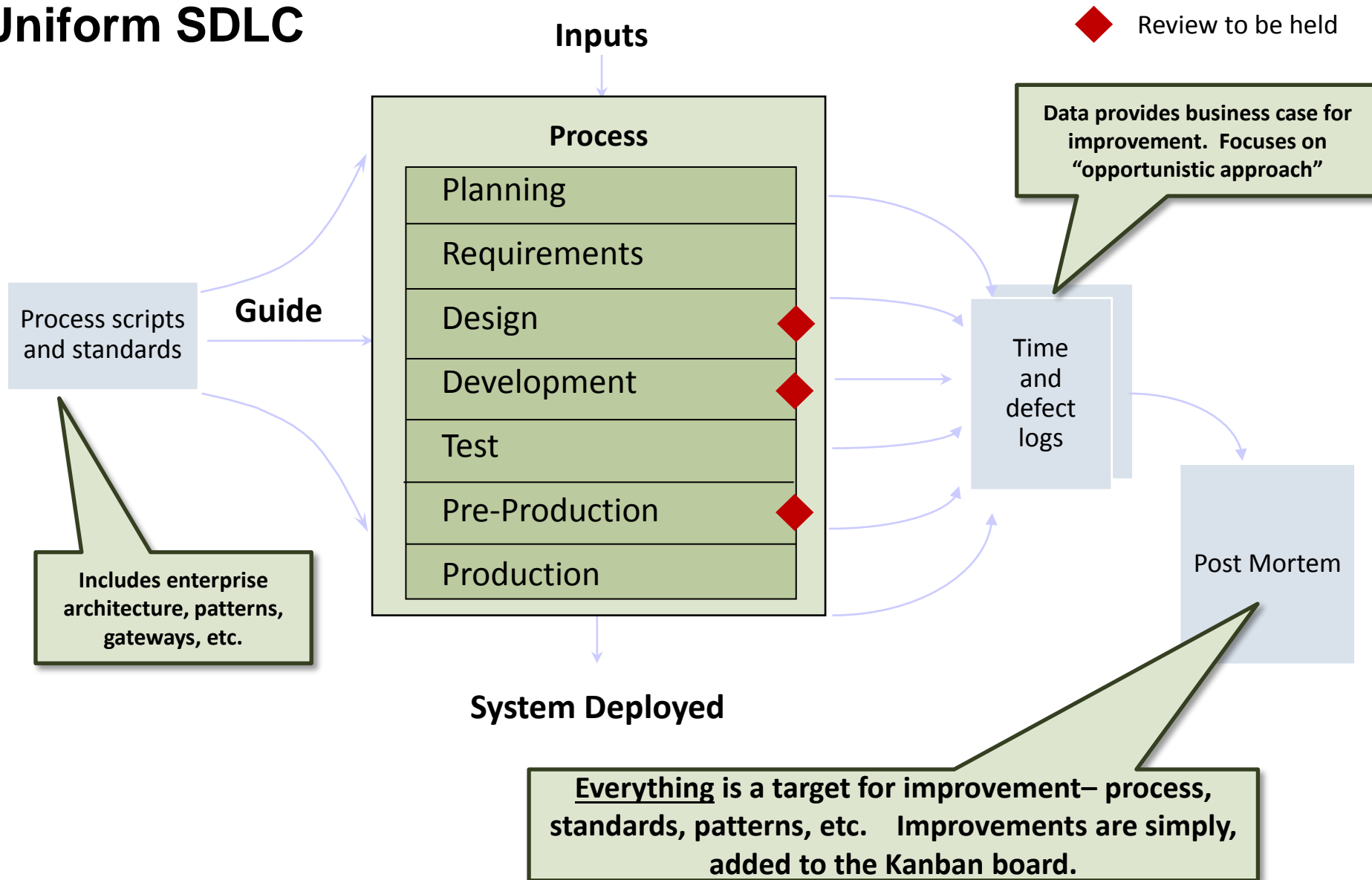
Through this effort, attention was captured among leadership. Software process and quality has become two leading objectives for ATK Enterprise.

1. A standard System Development Lifecycle is being promoted for each ATK location.

- allows for us to measure the process and improve the process by eliminate waste, reducing costs, reducing cycle times, etc.
- provides an opportunity for everyone to speak the same language
- improves consistency and our ability to share resources and technology.
- improves compliance with internal standards and regulatory compliance.
- improves reuse across organization

2. Focus on build in quality has become a key topic and organizations are beginning to focus on quality of execution rather than heroics.

Uniform SDLC



Process Scripts

2.1 SYSTEM DEVELOPMENT LIFE CYCLE PLANNING

SDLC planning provides an initial awareness for the request and allows the developer or solution architect to begin to plan the solution architecture to begin to plan the solution architecture. During this phase, the development planning process to support the request and high-level budgetary analysis.

2.2 PRE-DEVELOPMENT PREP

PRE-DEVELOPMENT PREP

	R	Application requirements required by the project.
	O	Project charter and high-level requirements.
	R	Resources for the project.
🔧	R	Ensure the application activity with the project.
	R	Identify which organization.
🔧	R	Ensure application requirements.

Standards

Pattern Overview

Pattern Name	ETL-FANOUT(Extract Transform Load with Fan Out Operation)
Pattern Description	In this pattern, data must be loaded into destination systems. The data is necessary for logging, persistence, configuration data will define the data.
Diagram	<pre> graph LR subgraph EndpointBox [Endpoint] Table end Trigger --> BlueBox[] </pre>

Design Requirements

Operating Type	This is the method to which it will run: application, unattended service, consumable service.
Technology Accepted	
Staging Data	
Configuration Data	If configuration data is required for an interface, where should it be stored.
Security	
Hard Errors	Critical errors trapped or untrapped that effect running of service.
Soft Errors	Errors handled via code within application.
Logging	Identifies what type of information should be logged regarding the running of the interfaces. This should include warnings, run stats, etc.
Trigger Mechanism	

Supporting Checklists

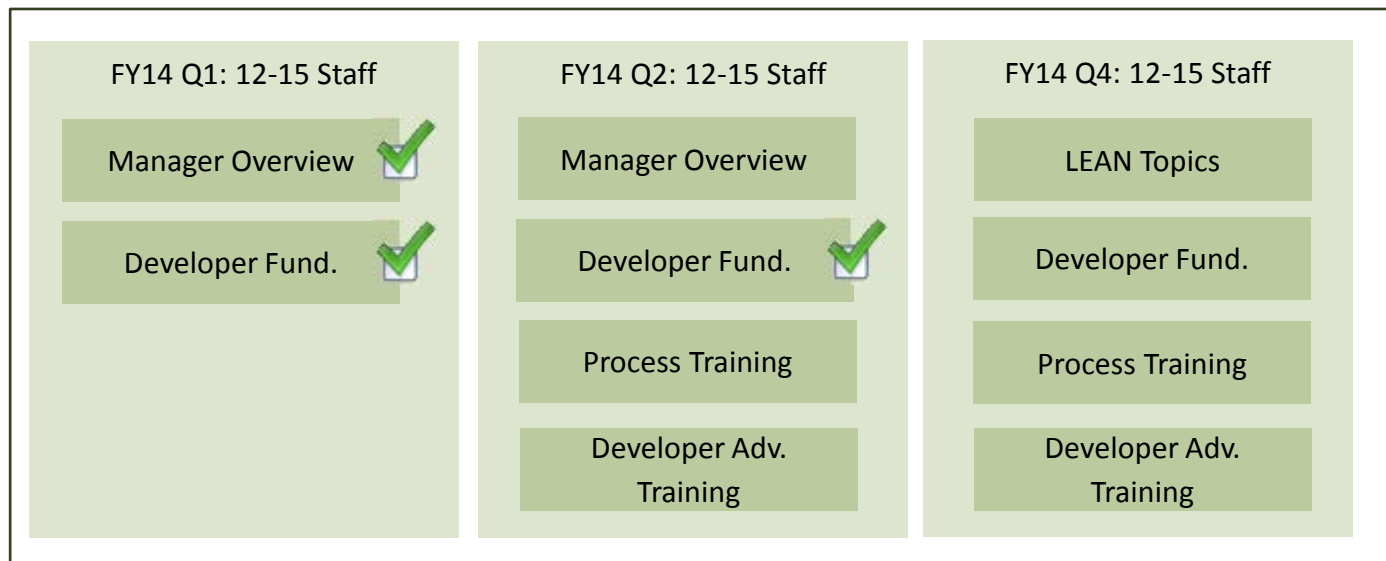
Requirements Definition

	Standard Activity	Local Process Step	Local Link / Template
✦ R	Work with requester and community of interest team to establish requirements for the request. Create a requirements statement and/or documentation.		TEMPLATE SAMPLE
R	Request existing designs, process diagrams and test plans for review		
O	Begin construction of Acceptance Test scenarios.		
R	Post documents to team location (Sharepoint, Subversion, etc.) as defined by governing organization.		SITE

Design Phase Activities

	Standard Activity	Local Process Step	Local Link / Template
R	Review guidelines and standards for system design, including security, standard software, etc.		Standards
R	Update design diagrams that include: proposed architecture, proposed function / procedures, proposed physical diagrams.		SAMPLE

- Training completed in three cycles. early adopters are targeted in the first group.
- All training includes manager kickoff meetings and follow ups.



- Second set of training incorporates “process training”.
- Coaching is to be provided to ensure practices are imbedded into organizations.

The PSP Fundamentals coursework:

- Provided a forum to talk about and discuss key issues, including priorities, capacity, planning, scheduling, etc.
- Able to de-sensitize resources of key issues, including defect tracking, post-mortems.

Adoptions:

- I went into it with a negative view but came away with my eyes opened. I'm not mandating we follow the PSP teachings as I want them both to come to their own conclusion.
- I have started a defect log and am updating my checklist. It is difficult as it is so easy to let the compiler find issues.
- We have established a set of project metrics that show the health of the process. This includes durations of schedules, limited tasks to maximum of one week of duration. Projects must be closed out, including lessons learned.
- We have started defect tracking – if a user reports a defect, we pretend we don't know it exists if we don't have it listed. Funny thing happened – the project manager started using our defect list to manage vendor relationships.

We focused on one problem at a time. Every 4-6 weeks, the problem changed. This required different techniques for addressing.

To many projects, constant changing priorities, unclear on what was next, and where things work.

Priority Management

To much assigned to one person, projects pushed onto teams.

Kanban

A lot of waste in the process, requirements changes, poor testing, etc.

Defect Tracking

Long period of time in project chartering phase, initial “go do” and kickoff with schedule earnest activity.

Process Scripts

Need to embed enterprise architecture and secure coding into the SDLC

Standards

Proactively address issues with workload, performance, etc.

Metrics

Thank You.

Prioritize Backlog



Align priorities with organization.

- Projects were scored to identify relative opportunities for action.
- Prioritized list was communicated with staff and management.
- Kept it simple – used Excel to establish process.

Benefits

- Priority was no longer questioned.
- Staff could see “work completed” over year.

ROW= Right of Way

Mgmt Rank	Scored Rank	Score	Project	Type	Date Requested	Scoring							
						+2	+2	+1	+10	+2	+2	+1	+2
						Process Improvement	Opp Cost	EADT	Direct	IWO	Audit / Legal	Run	Grow
ROW	1	11.5	Team Center Interfaces	Project	06/01/11	2	0	0	0	0	0	0	0
ROW	2	7	TEDS A3	A3	08/01/11	2	2	0	0	0	0	1	0
1	3	4	Application Deployment Technique	Project	09/01/11	2	0	0	0	0	0	0	2
2	6				09/01/11	2	0	0	0	0	0	1	0
3	7				09/01/11	2	0	0	0	0	0	0	1
4	4				9/1/2011	2	0	0	0	0	0	0	0
5	12				10/15/11	2	0	0	0	0	0	0	0
6	5				9/1/2011	2	0	0	0	0	0	0	0
7	8				08/01/11	0	0	0	0	0	0	0	0
8	9				9/1/2011	2	0	0	0	0	0	0	0
9	10				9/1/2011	2	0	0	0	0	0	0	0
10	11				9/1/2011	2	0	0	0	0	0	0	0
11	13				08/26/12	2	2	0	0	0	0	0	0
12	14				08/26/12	2	0	0	0	0	0	0	0
13	15	0	VOLT Views	SubProject	10/01/11	0	0	0	0	0	0	0	0
14	16	0	ERS Migration to DC	Task	09/01/11	0	0	0	0	0	0	0	0
15	17	0	Barcode Migration to DC	Task	09/01/11	0	0	0	0	0	0	0	0
16	18	0	PS Migration to DC	Task	09/01/11	0	0	0	0	0	0	0	0
17	19	0	SDLC Review	Task	10/15/11	0	0	0	0	0	0	0	0
18	20	3	Quality Environment Implementation	Project	08/26/12	2	0	0	0	0	0	0	0
19	21	-1	RFID Tool Load	Task	10/01/11	0	0	0	0	0	0	0	0
20	22	-1.5	SPSD PAC Changes	Task	10/05/11	0	0	0	0	0	0	0	0
21	23	-2	CP Architecture (Corp Review)	Task	10/15/11	0	0	0	0	0	0	0	0
22	24	1	MSG Settings & Err Handling	Project	08/26/12	2	0	0	0	0	0	0	0
23	25	1	Security Management - Applications	Task	08/26/12	2	0	0	0	0	0	0	0
24	26	-3	Improve Refresh Process	Task	10/15/11	2	0	0	0	0	0	0	0

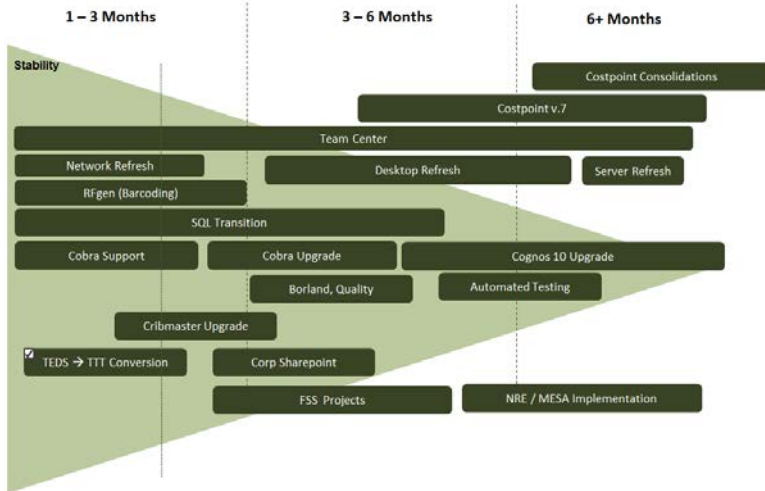
Rank includes:
Management Rank
Scored Rank

Scored by:
Process improvement
Opportunity Cost
Direct
Audit/Compliance
Run/Grow/Transform
Wait Time
Internal

Created a management oversight system



Business Systems "Core" Projects

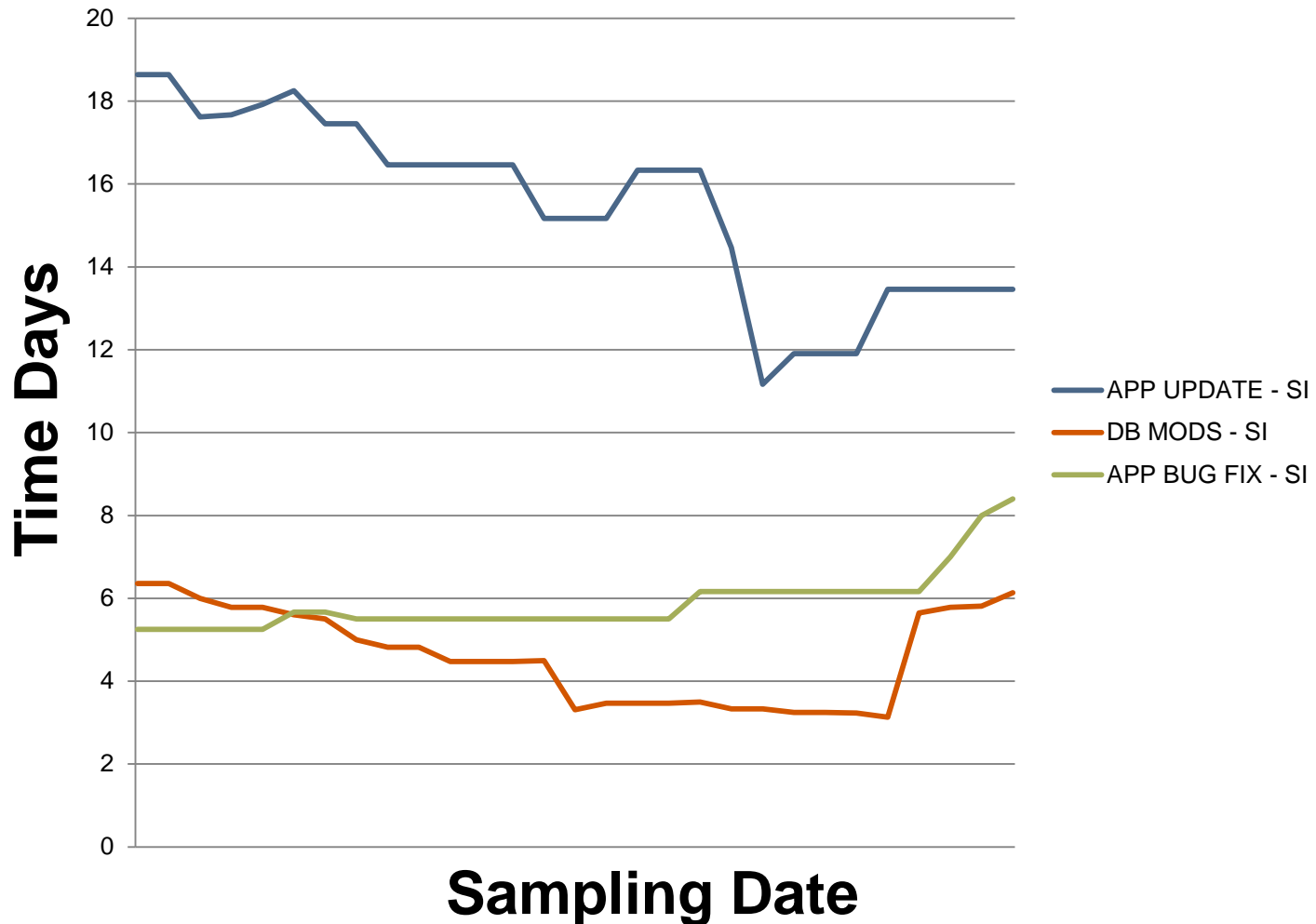


1. Monthly Step-Back reviews.
 - Provides broad picture, review execution against strategy.
2. Weekly Step-Back reviews.
 - Ensure monthly deliverables can be accomplished. Make changes as necessary.
3. Daily tiered meetings.
 - Issue resolution.
4. Bi-Weekly performance meetings.
5. Weekly Status reporting.

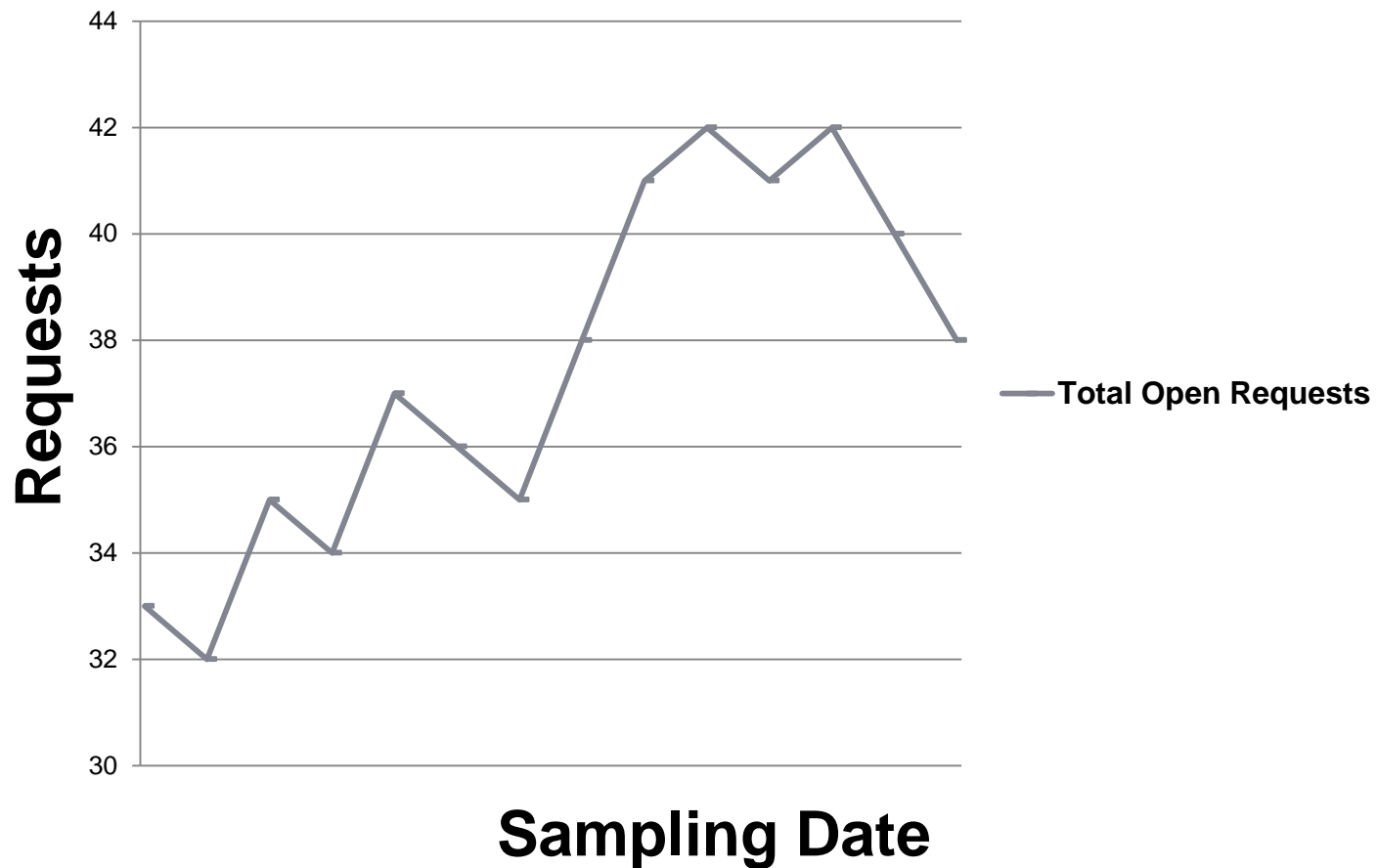
Wanted to ensure everyone knew what needed to be delivered, by when, and provided the tools to be successful.

	May	June	July	August								
Project	%Complete	%Complete	%Complete	FY13.5.1	FY13.5.2	FY13.5.3	FY13.5.4	FY13.5.5	Actual	Plan	Challenge	%Complete
TeamCenter	95.3%	72.7%	163.3%	1.5	7.25	5.25			14	30	16	46.7%
RFGEN	106.3%	187.5%	122.7%	1.5	4	5			10.5	16	5.5	65.6%
CobraSupport Upgrade	48.4%	118.1%	126.3%	11	21.5	40.25			72.75	80	7.25	90.9%
Network Refresh	33.3%	298.4%	100.0%	89	0	15			104	104	0	100.0%
Cribmaster	0.0%	175.0%	369.8%	4	7	1			12	20	8	60.0%
Cobra 5.1/WebPack/ WH Upgrades		99.2%	38.6%	2.25	1.75	38.75			42.75	40	-2.75	106.9%
Cash Receipts Interface		181.3%	96.9%	5.5	16.5				22	34	12	64.7%
Process Controllers			0.0%	0	0	0			0	0	0	
BurnR / BackFit Updates			0.0%	0	0	9			9	16	7	56.3%
Silk Test				0	6	6				16		

Cycle Times: We measure the time between the time start and the time finish on the respective activity.



Backlog: There is a direct correlation between compliance with reviews and the amount of items in our backlog.



Defect Yield: Track performance of team in reviews. Drop in performance is new resources being added to the process.

