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

<table>
<thead>
<tr>
<th>Soldier Systems</th>
<th>Rotary-Wing Military Aircraft</th>
<th>Fixed-Wing Military Aircraft</th>
<th>Commercial Aerospace</th>
<th>Satellites</th>
<th>Human Space Launch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Operations Forces</td>
<td>Sport Shooting</td>
<td>Law Enforcement</td>
<td>Ground Combat Vehicles</td>
<td>Naval Platforms</td>
<td>Satellite and Strategic Launch</td>
</tr>
</tbody>
</table>
ATK at a Glance

- 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
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

Utah
Brigham City
Clearfield
Logan
Magna

Virginia
Arlington
Hampton
Newington
Norfolk
Radford

West Virginia
Rocket Center

Dominican Republic
Santo Domingo

Puerto Rico
Lares
Mayaguez

Corporate headquarters and offices
Arlington, VA
Eden Prairie, MN

Sporting Group headquarters
Anoka, MN

Defense Group headquarters
Baltimore, MD

Aerospace Group headquarters
Magna, UT
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.
The early days...

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

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.
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.

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.

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

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

<table>
<thead>
<tr>
<th>Requirement / Action</th>
<th>Initial State</th>
<th>Current State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Projects in Team.</td>
<td>15+</td>
<td>6</td>
</tr>
<tr>
<td>Average Stop Work Issue Resolution</td>
<td>5-10 Days</td>
<td>1-2 Days</td>
</tr>
<tr>
<td>Cycle Time</td>
<td>10-12 Weeks</td>
<td>3 Weeks</td>
</tr>
<tr>
<td>Time on Task</td>
<td>9hrs p/week</td>
<td>12hrs p/week</td>
</tr>
</tbody>
</table>
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.

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.

Schedule Impact to Project (Days)

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.

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Activities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Establish Test Team</td>
<td>○ 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.</td>
</tr>
<tr>
<td>2</td>
<td>Deployment</td>
<td>○ Deploy the Application into the Testing Environment</td>
</tr>
<tr>
<td>3</td>
<td>Schedule Meeting</td>
<td>○ Each site that will use the application will need to be included on the required attendee list. ○ Each site will require a SME and tester(s). ○ 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. ○ If available, provide a user guide or help pages to assist in allowing the testers to understand how to use the application in question.</td>
</tr>
<tr>
<td>4a</td>
<td>Conduct Meeting</td>
<td>○ Provide an introduction with the purpose of the meeting</td>
</tr>
</tbody>
</table>
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!
How do we know it’s working?

• 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.
Where do we go from here…

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**
- Planning
- Requirements
- Design
- Development
- Test
- Pre-Production
- Production

**Inputs**

**Guide**
- Process scripts and standards
  - Includes enterprise architecture, patterns, gateways, etc.

**Time and defect logs**

**Review to be held**

**Data provides business case for improvement. Focuses on “opportunistic approach”**

**System Deployed**

*Everything is a target for improvement– process, standards, patterns, etc. Improvements are simply added to the Kanban board.*
### Process Scripts

#### 2.1 System Development Life Cycle Planning

SDLC planning provides an initial framework for the project and allows the developer to identify critical success factors. This phase includes identifying the problem, analyzing the requirements, and developing a plan.

#### 2.2 Pre-Development Preparation

- **R**: Application requirements as required by the organization.
- **C**: Project chart of work and high-level requirements.
- **R**: Resources for the project.
- **R**: Ensure the project is aligned with the organization.
- **R**: Ensure application requirements are met.

### Standards

#### Pattern Overview

**Pattern Name**: ETL-FANOUT (Extract, Transform, Load with Fan Out Operation)

**Pattern Description**: In this pattern, data must be distributed to multiple destination systems. The details necessary for logging, general configuration, and data will differ.

### Supporting Checklists

#### Requirements Definition

<table>
<thead>
<tr>
<th>Standard Activity</th>
<th>Local Process Step</th>
<th>Local Link / Template</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R</strong> Work with requester and community of interest to create requirements statements.</td>
<td></td>
<td>Template Sample</td>
</tr>
<tr>
<td><strong>R</strong> Review existing designs, process diagrams, and test plans for review.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>O</strong> Begin construction of Acceptance Test scenarios.</td>
<td></td>
<td>Site</td>
</tr>
<tr>
<td><strong>R</strong> Post documents to team location (Sharepoint, Subversion, etc.) as defined by governing organization.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Design Phase Activities

<table>
<thead>
<tr>
<th>Standard Activity</th>
<th>Local Process Step</th>
<th>Local Link / Template</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R</strong> Review guidelines and standards for system design, including security, standard software, etc.</td>
<td></td>
<td>Standards</td>
</tr>
<tr>
<td><strong>R</strong> Update design diagrams that include proposed architecture, proposed function / procedures, proposed physical diagrams.</td>
<td></td>
<td>Sample</td>
</tr>
</tbody>
</table>
Training Offering

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

### FY14 Q1: 12-15 Staff
- Manager Overview
- Developer Fund.

### FY14 Q2: 12-15 Staff
- Manager Overview
- Developer Fund.
- Process Training
- Developer Adv. Training

### FY14 Q4: 12-15 Staff
- LEAN Topics
- Developer Fund.
- Process Training
- Developer Adv. Training

- 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.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>To many projects, constant changing priorities, unclear on what was</td>
<td>Priority Management</td>
</tr>
<tr>
<td>next, and where things work.</td>
<td>Kanban</td>
</tr>
<tr>
<td>To much assigned to one person, projects pushed onto teams.</td>
<td>Defect Tracking</td>
</tr>
<tr>
<td>A lot of waste in the process, requirements changes, poor testing,</td>
<td>Process Scripts</td>
</tr>
<tr>
<td>etc.</td>
<td>Standards</td>
</tr>
<tr>
<td>Long period of time in project chartering phase, initial “go do”</td>
<td>Metrics</td>
</tr>
<tr>
<td>and kickoff with schedule earnest activity.</td>
<td></td>
</tr>
<tr>
<td>Need to embed enterprise architecture and secure coding into the</td>
<td></td>
</tr>
<tr>
<td>SDLC</td>
<td></td>
</tr>
<tr>
<td>Proactively address issues with workload, performance, etc.</td>
<td></td>
</tr>
</tbody>
</table>
Thank You.
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.

**Rank includes:**
- Management Rank
- Scored Rank

**Scored by:**
- Process improvement
- Opportunity Cost
- Direct
- EADT
- IWO
- Audit/Compliance
- Run
- Grow
- Transform
- Wait Time
- Internal
Created a management oversight system

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.


5. Weekly Status reporting.

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