Industry Recommendations for Implementing Continuous Iterative Software Development in the Defense Industry

NDIA Systems Engineering Division
in partnership with INCOSE and PSM

24-Oct-2019
Background

Defense Science Board (DSB) released a report in Feb-2018 containing seven recommendations regarding software design and acquisition. Section 868 of NDAA 2019 mandates implementation of these recommendations within 18 months.

The Defense Innovation Board (DIB) Software Acquisition and Practices (SWAP) study group has also provided many insightful and largely compatible recommendations.

NDIA, INCOSE and PSM support the DSB and DIB concepts and the opportunities they offer to DoD and the defense industry.

- NDIA offered the recommendations herein to ASD(A&S) and ASD(R&E) representing an “industry perspective” on path forward.
- NDIA appreciates the opportunity to partner with DoD on implementation.
DSB SW Task Force Recommendations

1. **Software Factory** – A key evaluation criteria in the source selection process should be efficacy of the offeror’s software factory.

2. **Continuous Iterative Development** – DoD and defense industrial base partners should adopt continuous iterative development best practices for software, including through sustainment.

3. **Risk Reduction and Metrics for New Programs** – For all new programs, starting immediately, implement best practices in formal program acquisition strategies (multiple vendors and down-selects, modernized cost and schedule measures, status estimation framework)

4. **Current and Legacy Programs in Development, Production, and Sustainment** – for ongoing development programs, PMs/PEOs should plan transition to a software factory and continuous iterative development.

5. **Workforce** – The U.S. Government does not have modern software development expertise in its program offices or the broader functional acquisition workforce. This requires Congressional engagement and significant investment immediately.

6. **Software is Immortal: Software Sustainment** – RFPs should specify the basic elements of the software framework supporting the software factory... reflected in source selection criteria

7. **IV&V for Machine Learning** – Machine learning is an increasingly important component of a broad range of defense systems, including autonomous systems, and will further complicate the challenges of software acquisition.

The NDIA working group developed consensus recommendations responding to each of the 7 DSB findings:

- Assumptions
- Picture of Success (End State)
- Current State
- Description
- Obstacles
- Path Forward

This briefing is an executive summary of those recommendations. Detailed report provided separately.
Framing Assumptions

Continuous iterative development (CID) methods have cross-functional implications. The scope includes not just SOFTWARE but also SYSTEMS ENGINEERING and supporting disciplines.

Software Factories include people, processes, and tools – not just a tool chain.

Funding and contracts must be aligned to support implementation and/or migration to SW factories with life cycle sustainment.

A collaborative approach to Intellectual Property (IP) across the entire acquisition life cycle will be developed that meets both Government and Supplier needs.

A business case can be made for the effective deployment and maintenance of integrated tool chains to build capability throughout the life of the system.

Traditional waterfall-based processes, tools, and measures are generally not well suited to CID.

A skilled SW-informed workforce cadre is available or can be developed across functions (e.g., software, acquisition, PMs, sustainment).

Cross-cutting assumptions. Refer to the separate detailed report with assumptions specific to each DSB recommendation area.
DSB #1: Software Factory
NDIA WG Recommendations

Picture of Success (end state)

People
- Qualified factory workforce
- Continuous learning (relentless improvement, pipeline feedback)

Process
- Integrated PMB
- Metrics
- Predictability
- Digital Blueprint / Play Book
- Ontology, Nomenclature
- Secure Supply Chain
- Relentless Improvement

Tools
- Platform agnostic tool chain
- Adaptable to change
- Test automation at all levels
- Model-based SW validation vs. architecture
- Red team / Blue team factory

Recommendations for Path Forward:

Security integrated into factory workflows (DevSecOps)
DSB #2: Continuous Iterative Development (MVP)
NDIA WG Recommendations

**Picture of Success (end state)**

**Government / Contractor Interface**
- New programs defined by solution intent (CV-1)
- Contracts defined by evolutionary viability products (MVP/NVP)

**Contracting**
- Contract funding structure supports seamless capability evolution

**Stakeholders**
- Active engagement in CID lifecycle

**Design**
- Guided by MOSA

**IP**
- Government access to source code with negotiated IP protections

**Program Execution**

**People**
- Multi-discipline agile execution includes aligned milestones
- Direct user/developer interaction informs design (product owner)

**Process**
- Early SE > SW sequencing, refactoring, tools, environments

**Tools**
- Test automation accelerates delivery (rapid release, deployment)

Procurements based on iterative development of releases according to product capability thresholds

**Recommendations for Path Forward:**

CID: Continuous Iterative Development
FOC: Final Operating Capability
IOC: Initial Operating Capability
IP: Intellectual Property
MOSA: Modular Open Systems Architecture
MVP: Minimally Viable Product
NVP: Next Viable Product

10/24/2019
NDIA Continuous Iterative Development and Sustainment WG
## DSB #3a: Risk Reduction (Competitive Prototyping)
### NDIA WG Recommendations

### Picture of Success (end state)

<table>
<thead>
<tr>
<th>Competition</th>
<th>Business case: win-win partnership, common goals, acquisition/support strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Objective downselect evaluation criteria (RFP L&amp;M) and feedback</td>
</tr>
<tr>
<td></td>
<td>Open architecture on critical components</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contracts</th>
<th>IP agreement negotiated, sustained across the life cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Funding and contracts aligned to support factory migration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Continuous improvement, SMART measures against objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Risk-based decision making</td>
</tr>
</tbody>
</table>

| Resources   | Funding, staffing, tools, environments to support multiple teams              |

### Recommendations for Path Forward:

**DSB #3a: Competitive Prototyping (1 of 2)**

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment strategy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Competition</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Cultural shift</td>
<td></td>
</tr>
<tr>
<td>Resources</td>
<td></td>
</tr>
<tr>
<td>Workforce development</td>
<td></td>
</tr>
</tbody>
</table>

**DSB #3a: Competitive Prototyping (2 of 2)**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Action Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td></td>
</tr>
<tr>
<td>measurements</td>
<td></td>
</tr>
<tr>
<td>IP strategy</td>
<td></td>
</tr>
</tbody>
</table>

**Competitive prototyping can help in many situations, but does not solve all acquisition problems.**

**NDIA Continuous Iterative Development and Sustainment WG**

10/24/2019
### DSB #3b: Measures for CID
NDIA WG Recommendations

<table>
<thead>
<tr>
<th>Picture of Success (end state)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Consensus frameworks</strong></td>
</tr>
<tr>
<td>• Objectives first - measures aligned and tailored from information needs, goals and constraints, at program and enterprise levels</td>
</tr>
<tr>
<td><strong>Modernized measures</strong></td>
</tr>
<tr>
<td>• Migration toward consensus alternatives to traditional waterfall and phase-based SW measures (LOC, EVM, milestones, ...)</td>
</tr>
<tr>
<td>• Derived from SW factory processes, automated by toolchain</td>
</tr>
<tr>
<td>• Basis for measuring cost and schedule vs. plan</td>
</tr>
<tr>
<td><strong>History-based estimates</strong></td>
</tr>
<tr>
<td>• Repositories collect performance-based measures (e.g., WBS, staff, cost, productivity) supporting future comparisons, basis of estimates, proposals, and program monitoring</td>
</tr>
</tbody>
</table>

Measures for CID should be aligned with information needs and constraints, at program and enterprise levels.

#### Recommendations for Path Forward:

<table>
<thead>
<tr>
<th>Software measurement framework for CID</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Use measure suite framework (objectives, categories, measures) with (EVM and industry standards - e.g., MIL-STD-125C, PSM, PMI)</td>
</tr>
<tr>
<td>• Further refines consensus measures for software CID</td>
</tr>
<tr>
<td>• History and validated measures after selected CID/evolutions programs</td>
</tr>
<tr>
<td>• Develop contracting language mapping measurement and forecasting for future programs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WBS-based tracking of historical competitiveness for staff costs, productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Recommended DOD/NSF-based approach and history of measures to additional programs for cost program level and not cap (e.g., continuous software /disposal shortfall consistent data per vendor)</td>
</tr>
<tr>
<td>• Performance improvement and cost competitiveness</td>
</tr>
<tr>
<td>• Partner with independent cost estimate (ICI) group to receive feedback and compare to industry for new methods (e.g., R&amp;D)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Match consensus on cost and schedule measures vs. plan for software CID</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Consider alternatives to EVM for managing performance vs. plan</td>
</tr>
<tr>
<td>• Reinc EM agile studies, publications, and guidance. Hold workshops with industry and developers to define framework, and measures.</td>
</tr>
<tr>
<td>• Reexamine consensus approach for CID software environment</td>
</tr>
</tbody>
</table>

CID: Continuous Iterative Development
EVM: Earned Value Management System
LOC: Lines of Code
WBS: Work Breakdown Structure
### DSB #4: Transition for Current and Legacy Programs
NDIA WG Recommendations

<table>
<thead>
<tr>
<th>Picture of Success (end state)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>People</strong></td>
</tr>
<tr>
<td>• Skill assessment for gap analysis</td>
</tr>
<tr>
<td><strong>Process</strong></td>
</tr>
<tr>
<td>• Business case for transition</td>
</tr>
<tr>
<td>• Playbooks and Blueprints for legacy code transition</td>
</tr>
<tr>
<td>• Assessment of supply chain and SW pedigree (FOSS, COTS, GOTS)</td>
</tr>
<tr>
<td>• Risk adjusted product backlog</td>
</tr>
<tr>
<td>• Strategies for incrementally building up test automation</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
</tr>
<tr>
<td>• Tools to generate legacy ‘as-built’ documentation and models for legacy code base</td>
</tr>
</tbody>
</table>

**Recommendations for Path Forward:**

See also: Defense Innovation Board SWAP Study Report: Supplementary Documents, Appendix B.6 Sustainment / Modernization Subgroup Report

**NDIA Continuous Iterative Development and Sustainment WG**

Click thumbnails to zoom
### DSB #5: Workforce
NDIA WG Recommendations

#### Picture of Success (end state)

<table>
<thead>
<tr>
<th>Education and Training</th>
<th>Stakeholder Engagement</th>
<th>Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• DAU curriculum for DevSecOps and modern SW-centric systems</td>
<td>• Collaborative government / industry partnerships</td>
<td>• Increased hiring, retention, training for acquisition experts</td>
</tr>
<tr>
<td>• Community of practice platforms</td>
<td>• Trained experienced industry partners and supply chain</td>
<td>• Recruiting pipeline for SW experts</td>
</tr>
<tr>
<td>• Training across career fields (PM, sustainment, acquisition)</td>
<td>• Consensus measurement framework</td>
<td>• Dedicated workforce funding and coaches across services</td>
</tr>
<tr>
<td>• Aligned with current/future development and recruiting needs</td>
<td>• Multi-discipline CID support teams (CDRLs, events, milestones)</td>
<td>• PMO IPTs for modern SW practices</td>
</tr>
</tbody>
</table>

#### Recommendations for Path Forward:

- Continuous Iterative Software Development, Acquisition and Sustainment
- DevSecOps
- Competency Models
- Practical Experience (Boot Camp)

10/24/2019

NDIA Continuous Iterative Development and Sustainment WG
DSB #6: Sustainment (Software Is Immortal)
NDIA WG Recommendations

### Picture of Success (end state)

<table>
<thead>
<tr>
<th>Resources</th>
<th>Availability and support of a trained proficient workforce</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Organic DoD software infrastructure, incentives, funding</td>
</tr>
<tr>
<td></td>
<td>Collaborative IP strategy throughout the life cycle, using a</td>
</tr>
<tr>
<td></td>
<td>“work shared sustainment” approach</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contracting Language</th>
<th>Contracts specify elements of framework supporting SW factory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Policies and guidance validated by workshops, pilots</td>
</tr>
</tbody>
</table>

| Sustainment Ecosystems         | Understanding of current and future organizational ecosystems |
|                                | to ensure effective transfer of SW factories                 |

### Recommendations for Path Forward:

1. Develop contracting language that contains the basic elements of the software framework supporting the software factory.
   - NDIA workshop with government and supplier personnel.
   - Generation and standardization of proposed contracting language.
   - Conduct a set of pilot programs.
   - Develop policies and guidance.

2. Develop an understanding of the current and future sustainment organizational components to ensure the effective transfer of the software factories.
   - NDIA workshop with government and contractor personnel.
   - Generation and standardization of effective transfer mechanisms.
   - Conduct a set of pilot programs.
   - Develop policies and guidance.

**NDIA Continuous Iterative Development and Sustainment WG**
DSB #7: IV&V for Machine Learning
NDIA WG Recommendations

**Picture of Success (end state)**

**Consensus ML IV&V Framework**
- Model-based inference engine considering full system context
- Risk-based methodology supporting T&E needs, linked to ML model failures early in system development process
- Mitigation throughout system design, development, sustainment

**Open Data Sets**
- High data quality, quantity, availability, and traceability
- Data repository accessible to government and industry
- Governance model for availability, level playing field, innovation
- New repository data continuously collected and published

**Perpetual Updates**
- Continuous ML model updates – evolution at speed of relevance
- Continuous V&V methods sensing changes from models, environment
- Performance/accuracy aligned with changing environment, threats

**Recommendations for Path Forward:**

**DSR #7: IV&V for Machine Learning (1 of 2)**
NDIA WG Recommendations

- **Innovative:**
  - Adopt a risk-based framework
  - Research and experimentation programs should place a primary focus on approaches to mitigate risks
  - Address ML risks/concerns within CDMOPs and architecture

- **Action Plan:**
  - Develop a risk management strategy for managing ML risk in the same way that cyber risk is managed
  - For the NDIA needs associated with ML in the system, use the mitigation of associated risks as a component of the test and evaluation process
  - Test ML programs against frameworks such as:
    - Data quality techniques to assess if training data sufficiently represent real-world data
    - Bias and Bias (AAT) approaches
    - Model and model integrity (e.g., through contract compliance of ML models)
    - Simulating trust in ML systems (see EIA-744, Reusable at (X))
    - Assess the extent to which a variety of ML methods are appropriate for a given context of ML models
    - Assess the extent to which a variety of ML methods are appropriate for a given context of ML models
    - Develop guidelines for the development and deployment of ML systems and models, and maintain data and model traceability

**DSR #7: IV&V for Machine Learning (2 of 2)**
NDIA WG Recommendations

- **Innovative:**
  - Ensure data availability and traceability across industry
  - Software Factory considerations for ML systems

- **Action Plan:**
  - Develop and maintain data exchange that meets the needs of a single reporting framework for data. This should include agreement on models and necessary security controls
  - Ensure all data generated by NDIA systems and deployed in systems for evaluation and deployment are shared, merged, and made available (if necessary) for ML and T&E
  - Establish criteria for a “Software Factory” that considers the special needs of ML systems
  - Evaluation criteria for Software Factory must consider the special needs of ML systems and should be updated as needed
  - Software Factory considerations include:
    - Limited access to training data (if available), e.g., restricted access
    - Limited access to training data (if available), e.g., restricted access
    - Limited access to training data (if available), e.g., restricted access
Summary

The NDIA WG provides an industry perspective on picture of success, current state, obstacles and path forward for each DSB recommendation

<table>
<thead>
<tr>
<th>DSB Recommendation</th>
<th>NDIA “Path Forward” recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 – Software Factory</td>
<td>14 Contracting, funding, incentives, methods, security, supply chain, and measures</td>
</tr>
<tr>
<td>#2 – Continuous Iterative Development</td>
<td>3 Pilots and continuous improvement</td>
</tr>
<tr>
<td>#3 – Risk Reduction &amp; Metrics</td>
<td>10 Acquisition strategy, competitive prototyping, culture, workforce, IP, and measures</td>
</tr>
<tr>
<td>#4 – Legacy Systems</td>
<td>5 Assessments, supply chain, methods, tools, and modeling</td>
</tr>
<tr>
<td>#5 – Workforce Development</td>
<td>3 Competency models, workforce assessment, workforce management, and training</td>
</tr>
<tr>
<td>#6 – Sustainment</td>
<td>2 Contracting and industry-government transfer of sustainment responsibilities</td>
</tr>
<tr>
<td>#7 – Machine Learning</td>
<td>5 Risk, research, CONOPs, ML data, and Software Factory interactions</td>
</tr>
</tbody>
</table>

Details of each topic and recommendation are provided in the separate report.
Acknowledgments

The NDIA Systems Engineering Division and its partners, INCOSE and PSM, appreciate the opportunity to provide an industry perspective for advancing the use of iterative methods in defense software acquisition.

The defense industrial base embraces the opportunities offered by the DSB and DIB recommendations and looks forward to supporting the Department of Defense with implementation.

NDIA Continuous Iterative Development and Sustainment Working Group:

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>Name</th>
<th>Company</th>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joseph Elm</td>
<td>L3 Technologies</td>
<td>Lemonte Green</td>
<td>MDA</td>
<td>Mike Phillips</td>
<td>SEI</td>
</tr>
<tr>
<td>Geoff Draper</td>
<td>Harris</td>
<td>Brian Hann</td>
<td>SAIC</td>
<td>Geoff Pierce</td>
<td>NRO</td>
</tr>
<tr>
<td>James Belford</td>
<td>USAF STSC</td>
<td>Stephen Henry</td>
<td>DAU</td>
<td>Marilyn Pineda</td>
<td>Lockheed Martin</td>
</tr>
<tr>
<td>Dawn Beyer</td>
<td>Lockheed Martin</td>
<td>Paul Janusz</td>
<td>US Army RDEC</td>
<td>Garry Roedler</td>
<td>Lockheed Martin</td>
</tr>
<tr>
<td>Barry Boehm</td>
<td>USC</td>
<td>Suzette Johnson</td>
<td>Northrop Grumman</td>
<td>Heather Romero</td>
<td>Raytheon</td>
</tr>
<tr>
<td>Kevin Chapman</td>
<td>Harris</td>
<td>Cheryl Jones</td>
<td>US Army CCDC Armaments</td>
<td>Gene Rosenbluth</td>
<td>Northrop Grumman</td>
</tr>
<tr>
<td>Yann Chazal</td>
<td>Renault</td>
<td>Geethesh Kukkala</td>
<td>SAIC</td>
<td>Larri Rosser</td>
<td>Raytheon</td>
</tr>
<tr>
<td>David Chesbrough</td>
<td>NDIA</td>
<td>Richard Kutter</td>
<td>USAF</td>
<td>Dan Strickland</td>
<td>MDA</td>
</tr>
<tr>
<td>Chris Collins</td>
<td>DAU</td>
<td>John MacCarthy</td>
<td>Univ. of Maryland</td>
<td>James Thompson</td>
<td>OUSD(R&amp;E) retired</td>
</tr>
<tr>
<td>Mark Cornwell</td>
<td>OUSD(R&amp;E)</td>
<td>Phyllis Marbach</td>
<td>INCOSE</td>
<td>Steve Verga</td>
<td>Harris</td>
</tr>
<tr>
<td>Truc DeSa</td>
<td>Lockheed Martin</td>
<td>Jason McDonald</td>
<td>Harris</td>
<td>Ketchiozo Wandji</td>
<td>NAVAIR</td>
</tr>
<tr>
<td>James Dowse</td>
<td>US Army ARDEC</td>
<td>Mike McLendon</td>
<td>SEI</td>
<td>Allison Weigel</td>
<td>Toray</td>
</tr>
<tr>
<td>Rick Dove</td>
<td>Paradigm Shift</td>
<td>Jenna Meyers</td>
<td>HQDA ASA FM</td>
<td>Beth Wilson</td>
<td>retired</td>
</tr>
<tr>
<td>Jim Duffy</td>
<td>Raytheon</td>
<td>Jeffrey Mueller</td>
<td>DAU / USAF</td>
<td>Erik Wylie</td>
<td>MDA</td>
</tr>
<tr>
<td>Robert Epps</td>
<td>retired</td>
<td>Kenneth Nidiffer</td>
<td>SEI</td>
<td>Hasan Yasar</td>
<td>SEI/CERT</td>
</tr>
<tr>
<td>Mark Ginese</td>
<td>DAU</td>
<td>John Norton</td>
<td>Raytheon</td>
<td>Robin Yeman</td>
<td>Lockheed Martin</td>
</tr>
<tr>
<td>Firas Giaiel</td>
<td>Raytheon</td>
<td>Virginia Perkins</td>
<td>MDA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10/24/2019 NDIA Continuous Iterative Development and Sustainment WG 14
For More Information ...

Contact:
Robin Yeman
Lockheed Martin Corp.
robin.yeman@lmco.com

Joseph Elm
Elm System Solutions
jpelm1@consolidated.net
Backup

Supporting Content (Hidden Slides)

Excerpts of NDIA Recommendations by DSB Finding

(see separate briefing package for full details)
DSB #1: Securing the Factory

With security integrated

Product Backlog

Release Backlog

Automated Security tests
Run daily

Fuzzing

Plan
Formal Security Modeling

Design
Secure Design Principles

Build
Defensive Coding

Test
Self Service Security scanning

Definition Of Done (sprint)
Security checks

Fuzzing

Pen Test
Dynamic Analysis

Security checks

Definition Of Done (Release)

Security checks

Shippable Product

Continuous Monitoring

10/24/2019 NDIA Continuous Iterative Development and Sustainment WG
<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
</table>
| **Contract for software factory delivery** | • Create a blueprint of contracts and language to enable software factory delivery  
• Define approaches for different types of software (e.g., embedded, firmware, web); (life-critical, business-critical, low risk) |
| **Fund value streams instead of projects** | • Pilot funding a value stream for a single vendor award program  
• Pilot funding value streams on multi-vendor award program |
| **Incentivize Suppliers to build interoperable software factories that are continuously exercised** | • Hold workshop with Industry to identify incentives  
• Pilot options on some small short term modular contracts |
| **Standardize software factory interfaces to facilitate data sharing** | • Common data architecture  
• Define standards at the data layer for software factory to enable flexibility  
• Define common nomenclature standards across vendors; use an existing framework such as the Scaled Agile Framework (SAFe) |
## Initiative Action Plan

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publish blueprints and playbooks</td>
<td>• Collaborate with Industry to obtain software factory blueprints and playbooks and publish for use across programs to increase success</td>
</tr>
</tbody>
</table>
| Transparent integrated PMB | • Publish blueprint of Integrated PMB (may differ across domains)  
• Educate Government PMs on how to review PMB |
| Securing software factory | • Define a defense-in-depth approach to secure factory  
• Identify a required cadence of Red Team / Blue team to ensure factory safe. |
| Standards-based supply chain | • Define supply chain standards  
• Define interoperability for supply chain with multiple factories |
| Define value stream for delivery and push varied vendor baselines through factory | • Define value stream for delivery and enable multiple vendor baselines to deliver into the factory.  
• Ensure interoperability |
## DSB #1: Software Factory (3 of 3)
### NDIA WG Recommendations

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
</table>
| **Measure practices and process for results** | • Document program practices and processes being used  
• Measure success of programs by practice and environment to analyze which practices are demonstrating the best results based on customer criteria of value. (not methodology, but individual practice) |
| **DoD-run retrospectives for a sampling of programs** | • Select a sampling of programs once a quarter and run a retrospective jointly between Industry and Government to identify root causes and improvements  
• Publish best practices identified in retrospectives for all vendors |
| **Open source** | • Research approach to instantiate Government-based open-sourced ways of working to leverage common modules across vendors and programs |
| **Teams as a service (CID Cells)** | • Research approach to leverage cross-functional teams as a service in work areas were there is higher availability of workforce. |
| **IATO for infrastructure** | • Research opportunity to obtain IATO on Infrastructure of software Factory.  
• bare metal / cloud / database (DB) are the longest lead-time items to approve  
  o If we could secure a common architecture, the application layer would be cheaper and faster to approve, reducing cycle time for capabilities |
DSB #2: Continuous Iterative Development
Picture of Success (End State)

**Government/Contractor Interface:**

- New Programs Defined by Solution Intent
- Contracts Defined by Minimal Viable Product (MVP)
- Funding Supports Capability Evolution
- Stakeholders Actively Engaged in Continuous Iterative Development Lifecycle
- Design Guided by MOSA
- Government Access to Source Code with IP Protections

**Program Execution:**

- Multi-discipline agile execution includes milestones
- Direct user interaction informs design
- Test automation accelerates delivery
### Initiative: Establish CID pilot baseline
- Establish & communicate an initial high level CID approach
- Establish an initial approach to defining programs for CID implementation
- Train key Government and Supplier personnel

### Initiative: Pilot, learn and refine
- Define a design set for CID
- Conduct pilot programs for CID, employing a set based design approach to explore options and refine approach
- Iterate until a small set of effective approaches and techniques emerge and standardize on it

### Initiative: Implement and evolve
- Develop an approach to integrate feedback into the standard process for continuous improvement
- Define CID requirement phasing and Inspect and Adapt workshop timing
- Roll out CID as standard approach
- Manage feedback and evolution
Study indicates that CP can help in many situations, but has a number of pitfalls. CP does not solve all acquisition problems.
<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition strategy</td>
<td>Acquisition strategies that provide a fair opportunity to compete, retain competition throughout the lifecycle for critical components to enable rapid evolution of the product.</td>
</tr>
<tr>
<td>Competitive prototyping</td>
<td>• Review analyses/reports from prior DoD competitive prototyping initiatives, and integrate lessons learned into action plan for DSB recommendations.</td>
</tr>
<tr>
<td></td>
<td>• Competitive prototyping risk reduction strategy should account for both functional and non-functional requirements.</td>
</tr>
<tr>
<td>Cultural shift</td>
<td>Migrate from subjective qualitative assessment to objective quantitative assessment of risk that support business decisions</td>
</tr>
<tr>
<td>Resources</td>
<td>DoD investment to acquire, deploy, integrate, and maintain evaluation tools and test beds</td>
</tr>
<tr>
<td>Workforce development</td>
<td>Recommend DoD initiate a development plan to provide workforce with skills and knowledge needed to plan, perform and execute the risk reduction strategies during competitive prototyping.</td>
</tr>
</tbody>
</table>
### Initiative | Action Plan
--- | ---
**Program measurements** | • Define a minimum core set of metrics and ownership for measures needed to do the job at the Program, Functional, and Integrated Product Team (IPT) levels  
• Develop and track metrics to control factory processes, measure against goals and objectives, assess/measure risk, and make decisions  
• Enable real-time insight into measures and program status  
• Ensure measures provide a comprehensive view of risk reduction strategy, including: functional and non-functional requirements; reliability, security, ...  
• Develop consensus Government/Industry measurement framework and common measures applied across defense software acquisition programs.

**IP strategy** | • Develop contracting approaches that protect Supplier IP while providing the Government access to source code for analysis, deployment, support, and evolution.  
• Sustain IP required for maintenance of the following:  
  • Renewable capital – patents, license, IP, ...  
  • Human capital – People, skills, experience, surge/slack...  
  • Structural capital – data bases, tools, processes, test scripts, ...  
  • Relationship capital – customers, supplier agreements, business relationships, personal relationships, ...
DSB Recommendation #3 - Metrics
Frameworks for aligning measures with objectives

Measures for continuous iterative development should be aligned with information needs, objectives and constraints, at program and enterprise levels

Summary of DIB Metrics Categories

**Deployment Rate**
- Initial launch to deployment of simplest useful functionality [MVP]
- Time to field high priority fn (spec>ops) or security hole (find>ops)
- Time from code committed to code in use

**Response Rate**
- Time req'd for full regression test (automated) and cyber testing
- Time required to restore service after outage [MTTD, MTTR, MTTA]

**Code Quality**
- Automated test coverage of specs / code
- Number of bugs caught in testing vs. field use [defect detection %]
- Change failure rate (rollback)

**Program Management**
- Complexity metrics. Devel plan/env metrics (specs, code, staff, ...)

Measures, goals, and priorities are tailored based on program objectives and information needs

The NDIA WG recommends a measurement framework that can be adapted to specifics of the program, domain, or acquisition

Industry feedback
(usefulness, effectiveness)

Availability measures
instrumented and automated by the toolchain

Selection of program measures
tailored by information needs
(with a few primary colors required by the enterprise)

Enterprise measures
driven by business performance objectives

Success is measured at multiple levels:
- Mission capability
- Program execution
- Enterprise improvement
- Business results, competitiveness

 Adoption

NDIA Continuous Iterative Development and Sustainment WG
<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
</table>
| Software measurement framework for CID | • Validate measurement framework (objectives, categories, measures) with Government and industry stakeholders (e.g., NDIA, INCOSE, PSM, SERC)  
• Finalize initial consensus measures for software CID  
• Pilot and validate measures/analysis on selected CID /DevSecOps programs.  
• Develop contracting language requiring measurement set for future programs |
| WBS-based estimating of historical comparables for staff, cost, productivity | • Recommend DoD expand WBS-based approach and historical DB measures to additional programs but at program level and not specific to continuous software initiatives (doubtful consistent data yet exists).  
• Engage Government stakeholders on historical data estimating initiatives  
• Partner with independent cost estimate (ICE) groups to migrate away from SLOC-based methods (CAPE, PARCA, ICE, …); establish partnerships with industry for new methods (DSB #3) |
| Reach consensus on cost and schedule measures vs. plan for software CID | • Consider alternatives to EVM for managing performance vs. plan.  
• Review EVM agile studies, publications, and guidance. Hold workshops with Industry and Government to define framework and measures.  
• Recommend consensus approach for DoD software acquisition |
## DSB #4: Current and Legacy Programs
NDIA WG Recommendations

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
</table>
| Program assessment for categories of legacy software programs. | • Collaborate with industry building program categorization table for varied types of software and products being built  
• Define common list of program readiness attributes  
• Define metrics for how to measure transition success  
• Develop common risk categories to evaluate  
• Prototype process for iteratively and incrementally transitioning programs |
| Supply chain pedigree evaluation tool                          | • Investigate methods for evaluating software pedigree  
• Prototype process and tools to evaluate supply chain pedigree  
• Validate pedigree on FOSS/COTS/GOTS/Supplier components |
| Blueprints and playbooks for low risk transition               | Collaborate with Industry to build repository of blueprints, playbooks, and strategies for different types of programs.                                                                                     |
| Visualization tools for varied code bases.                     | Investigate Visualization tools for different types of code bases                                                                                                                                            |
| Auto generate “As-Built” and Models to evaluate system and develop transition plans | • Investigate standardized set of tools to auto-generate models and “As-Built” of the varied legacy systems  
• Define a prioritization strategy for migrating program components to the software factory |
# Initiative: Modern software-intensive-systems engineering competency model development

- DAU/INCOSE/NDIA/ISO collaboration to add software-centric systems engineering roles and proficiencies to INCOSE SE competency model and identify / develop workforce development content to improve proficiency.
- Create ability to ID/code software-intensive-systems engineering in current/future software-centric systems skillsets.

# Initiative: Informed PMs and software SMEs Training

- Development and deploy training at Defense Acquisition University on iterative software development for all acquisition communities (PM, Systems Engineering, Software, Financial Management, Cost Estimating, ...)
- Develop a consensus government/industry measurement framework and common measures applied across defense software acquisition programs.
- Supply chain integration - Deploy supply chain pedigree evaluation tools and techniques.
- Develop blueprints and playbooks for low risk transition.
- Develop RFP guide for acquiring and transitioning to software factories.
### Initiative: Workforce management

<table>
<thead>
<tr>
<th>Action Plan</th>
</tr>
</thead>
</table>
| • Baseline current software intensive capabilities and needs  
  • Identify workforce gaps; quantity/quality  
  • Update workforce needs to shape workforce recruitment and training  
  • Create a new software-centric-systems Engineering 0800 Occupational Series to enable tracking, management and growth of software-centric-systems engineers, managers, and functional personnel  
  • Fund software intensive develop training  
  • Support continuous learning |
## DSB #6: Software Sustainment
### NDIA WG Recommendations

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop contracting language that contains the basic elements of the</td>
<td>• NDIA workshop with government and Supplier personnel&lt;br&gt;• Generation and socialization of</td>
</tr>
<tr>
<td>software framework supporting the software factory</td>
<td>proposed contracting language&lt;br&gt;• Conduct a set of pilot programs&lt;br&gt;• Develop policies and</td>
</tr>
<tr>
<td></td>
<td>guidance</td>
</tr>
<tr>
<td>Develop an understanding of the current and future sustainment</td>
<td>• NDIA workshop with government and contractor personnel&lt;br&gt;• Generation and socialization of</td>
</tr>
<tr>
<td>organizational ecosystems to ensure the effective transfer of the software</td>
<td>effective transfer mechanism&lt;br&gt;• Conduct a set of pilot programs&lt;br&gt;• Develop policies and</td>
</tr>
<tr>
<td>factories.</td>
<td>guidance</td>
</tr>
<tr>
<td>Initiative</td>
<td>Action Plan</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **Adopt a risk-based framework** | Deploy a risk-based framework for managing ML risk in the same way that cyber risk is managed  
  • For the IV&V needs associated with ML in the system, use the mitigation of associated risks as a core part of the test and evaluation process |
| **Research and experimentation programs should place a primary focus on approaches to mitigate risks** | Pilot R&D programs focused on approaches such as:  
  • Data quality techniques to assess if training data sufficiently represent real-world distributions  
  • Run Time Assurance (RTA) approaches  
  • Formal methods and other approaches to prove correctness of ML models  
  • Enhancing trust in ML systems (see DARPA Explainable AI (XAI)) |
| **Address ML risks/concerns within CONOPS and architecture** | Standardize approaches to evaluating ML risk in the system, and develop playbook of, CONOPS, architectural frameworks, and design patterns to mitigate these types of risk  
  • The risks associated with ML in a system depends on how that ML model impacts overall system behavior  
  • We can manage risk levels through CONOPS and system architecture decisions |
### NDIA WG Recommendations

#### Initiative Action Plan

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Action Plan</th>
</tr>
</thead>
</table>
| Ensure data availability and traceability across industry | Establish a data exchange that is not just a simple repository/dumping ground for data... Instead espousing a governance model and necessary security controls  
- **DIB:** “All data generated by DoD systems - in development and deployment - should be stored, mined, and made available for machine learning (ML)”  
- To allow for greater innovation, make all this data available to industry via a secure data repository/exchange  
- Include requirements for maintaining history, provenance and pedigree of data sets and ML models, and maintain data/model traceability  
- Continuous V&V methods tied to sensing of changes from models & environment |

| Software factory considerations for ML systems | Ensure that evaluation criteria for a “Software Factory” considers the special needs of ML systems:  
- Evaluation criteria for Software Factories must consider the special needs of development and deployment for ML (models need to be rapidly re-trained, re-tested, re-deployed) Software factory considerations include: abundant storage for training/validation data, ample compute (e.g., Graphics Processing Units (GPUs), Tensor Processing Units (TPUs)) to support training runs, etc. |