NDIA

National Defense Industrial Association

Implementing Continuous Iterative Development and Acquisition

22-Apr-2019

Background

Defense Science Board (DSB) released a report in Feb-2018 containing seven recommendations regarding software design and acquisition

Section 868 of National Defense Authorization Act (NDAA) 2019 mandates implementation of these recommendations within 18 months

- The Undersecretary of Defense (Acquisition and Sustainment (USD(A&S)) has the lead on implementing recommendations 1 thru 6
- The Undersecretary of Defense (Research and Engineering) (USD(R&E)) has the lead on implementing recommendation 7

The National Defense Industrial Association (NDIA), in collaboration with the International Council on Systems Engineering (INCOSE) and Practical Software and Systems Measurement (PSM) has volunteered to provide input to USD(A&S) and USD(R&E) representing the "industry perspective" on implementation of the DSB recommendations

- While the DSB report focuses primarily on SOFTWARE design and acquisition using continuous and iterative methods, NDIA believes that the scope must be expanded to focus on SYSTEM design and acquisition using continuous and iterative methods.

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 (CID)** – The Department of Defense (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, Program Managers (PMs)/ Program Executive Officers (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** – Requests for Proposals (RFPs) should specify the basic elements of the software framework supporting the software factory … reflected in source selection criteria
7. **Independent Verification and Validation (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**

DSB #1: Software Factory

Software Factory

… establish a common list of source selection criteria for evaluating software factories for use throughout the Department. To be considered minimally viable for a proposal, competing Suppliers should have to demonstrate at least a pass-fail ability to construct a software factory.

DSB #1: Software Factory
Picture of Success (end-state) 1

Type	Success	Criteria	Verification Approach
People	**Qualified factory workforce**	Demonstrable skills and experience in software development for Industry and Government personnel.	Interview personnel who are supporting the software factory.
People	**Continuous learning**	Skill plans and paths for staff to level-up skills.	Review skill paths and associated training available.
Process	**Integrated performance measurement baseline (PMB).**	Soft link all of the tools in the value stream to deliver software.	Review that all of the tools are soft linked. • Requirements Tools • Product Backlog • Master Schedule • Models • Repository • Test Tools • Deployment Tools That demonstrates end-to-end traceability

DSB #1: Software Factory
Picture of Success (end-state) 2

Type	Success	Criteria	Verification Approach
Process	**Suggested metrics for software factory** • **Lead time** • **Cycle time** • **Mean time to restore** • **Mean time to attack** • **Mean time to detect**	The ability to demonstrate the system metrics for previous items that have moved through the software factory	Review **run charts** of past products that have been delivered through the software factory
Process	**Predictability**	Demonstrable past performance of software projects - planned vs. actual (scope and schedule)	Review past **run charts**
Process	**Digital blueprint / playbook repository**	Digital repository of blueprints and playbooks for the software factory that is searchable	Demonstrate search for software development practice or process
Process	**Common ontology and nomenclature**	Common ontology and nomenclature for process to enable communication	Review of ontology and nomenclature materials

DSB #1: Software Factory
Picture of Success (end-state) 3

Type	Success	Criteria	Verification Approach
Process	**Secure supply chain**	Provide ability to show Free and Open Source (FOSS)/Commercial Off-the-Shelf (COTS)/ Government Off-the-Shelf (GOTS)/Supplier pedigree in the tool chain or software deliverable	Review FOSS/COTS/GOTS/ Supplier products and a validated test reports.
Process	**Relentless improvement**	Process exist describing relentless improvement.	Review Process for relentless improvement.
Tools	**Fast feedback on system performance.**	System telemetry available real time on team monitors.	Review system telemetry from the software factory
Tools	**Existing integrated automated tool chain that is platform agnostic.**	Tool chain exists to perform continuous integration, continuous test, continuous deployment, system telemetry, etc. on day 1 of contract start.	Build "Hello World" capability in whatever language we are using and review successful: • Build results • Integration results • Static analysis results • Dynamic analysis results • Automated test results • System telemetry results • Deployment

DSB #1: Software Factory
Picture of Success (end-state) 4

Type	Success	Criteria	Verification Approach
Tools	**Test automation at all levels.**	Multi-tier automated test harness.	Review coverage of automated test • Unit • Integration • Performance • Security • Functional
Tools	**Software validation against the software architecture models.**	Validate all software built against the architecture using models.	Evaluation of software against the models to determine divergence from the architecture.

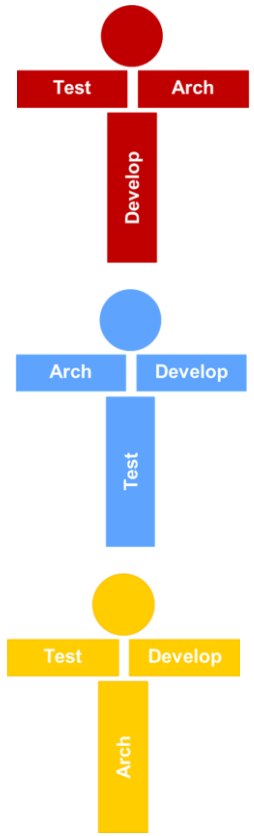
DSB #1: Software Factory
Picture of Success (end-state) 5

Type	Success	Criteria	Verification Approach
Tools	**The integrated toolchain for the software factory is highly cohesive and loosely coupled to enable change of tools easily.**	Integration at the data layer through Application Programming Interfaces (APIs) or data Attributes. (Tools such as *Tasktop* to integrate)	Review to ensure there is not any 'point-to-point' integration. Change one tool out for an equivalent one and verify that the factory runs the same and provides the same metrics.
Tools	**Red Team / Blue Team factory**	Verify security of the factory and products within the factory	Review Red Team results including metrics such as Mean Time To Attack (MTTA) and Mean Time to Detect (MTTD)

DSB #1: Software Factory Concept

All factories require people, process, and tools to run

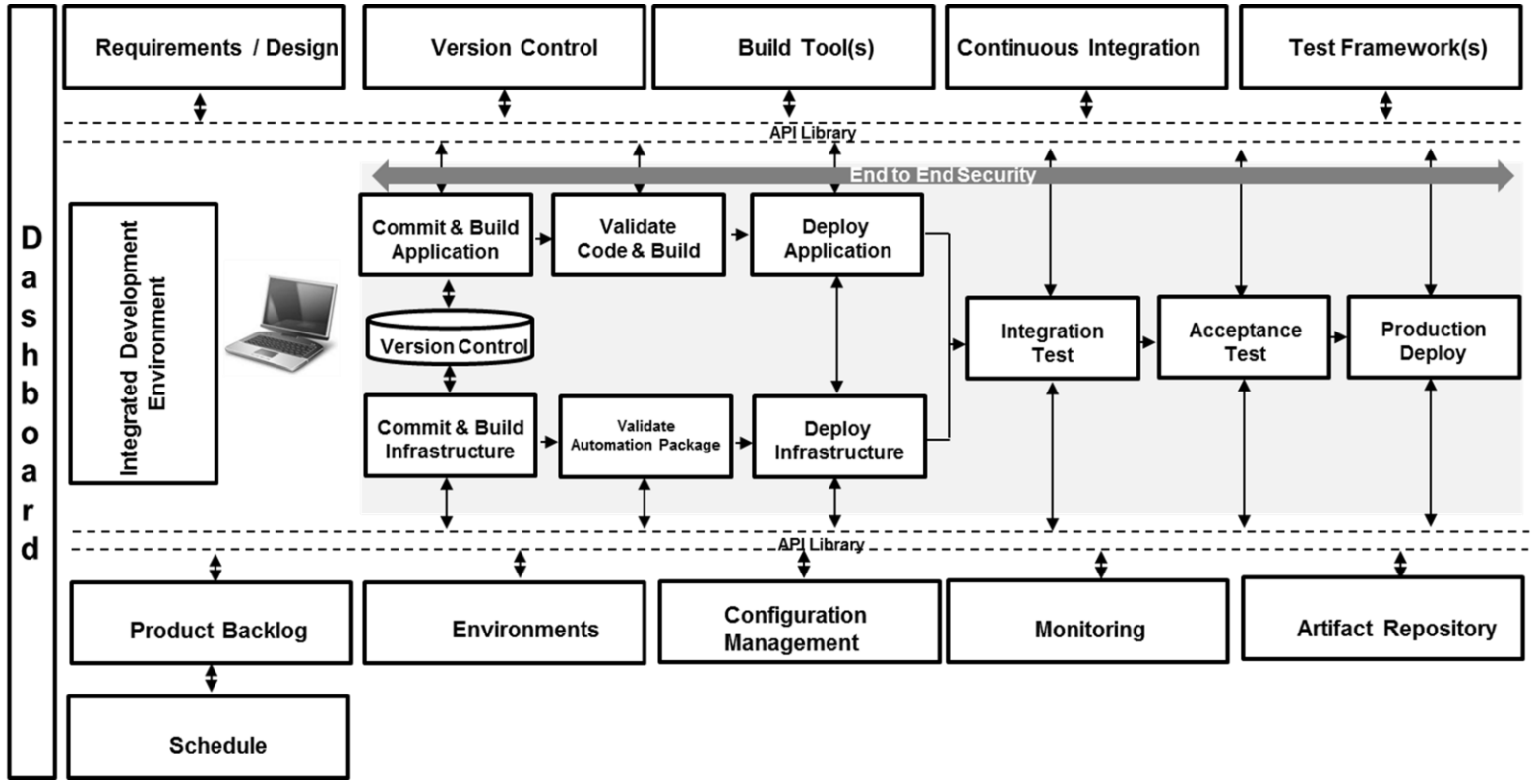
People



Process

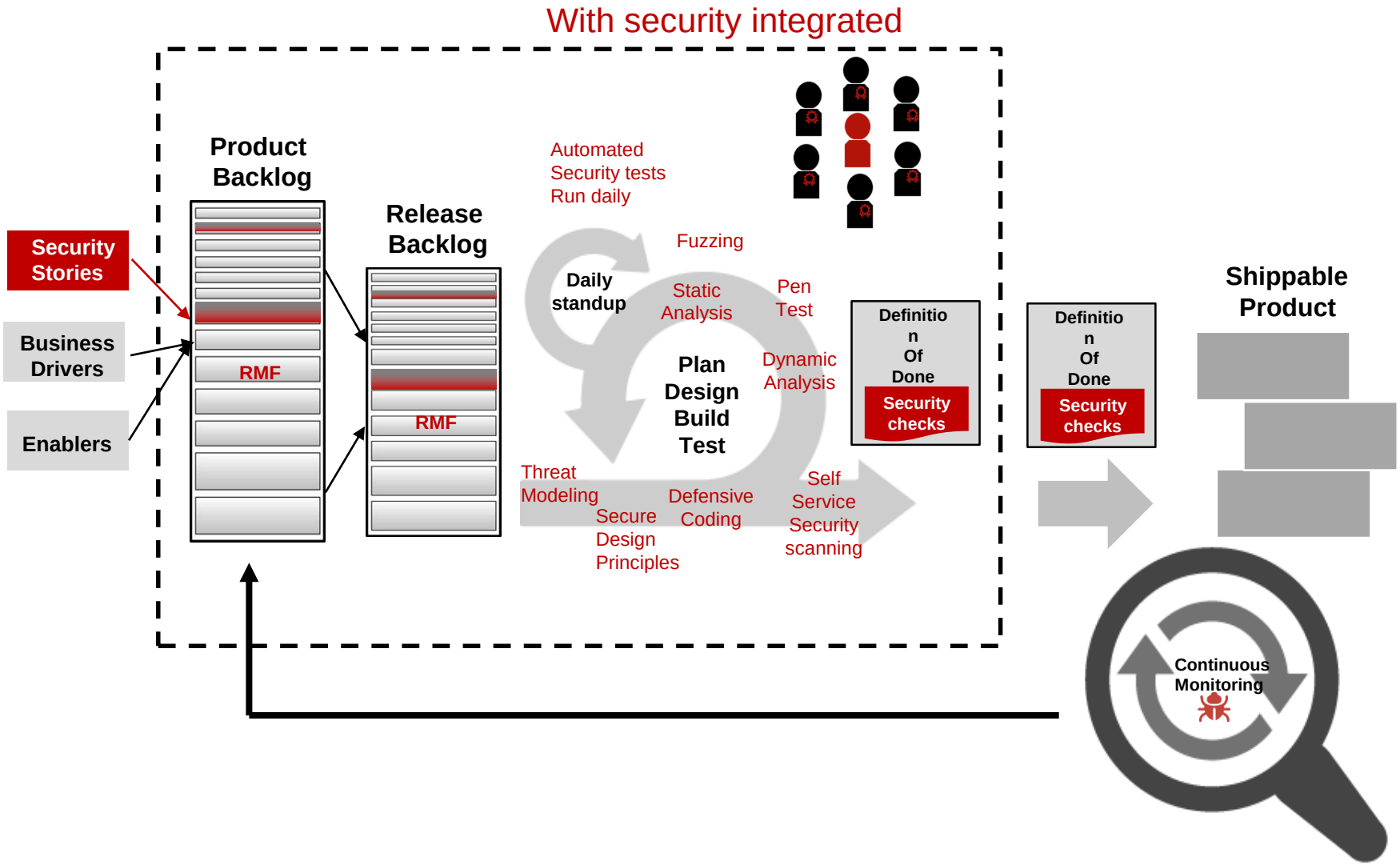
- Multiple Horizons of Planning
- Daily stand-up
- Heartbeat
- Time boxing
- Retrospective
- Product backlog
- User stories
- Version control
- Personas
- Pairing
- Collective ownership
- Backlog grooming
- Test-driven development
- Continuous integration
- Multi-tier automated test
- Visible progress indicators

Integrated Tool Chain



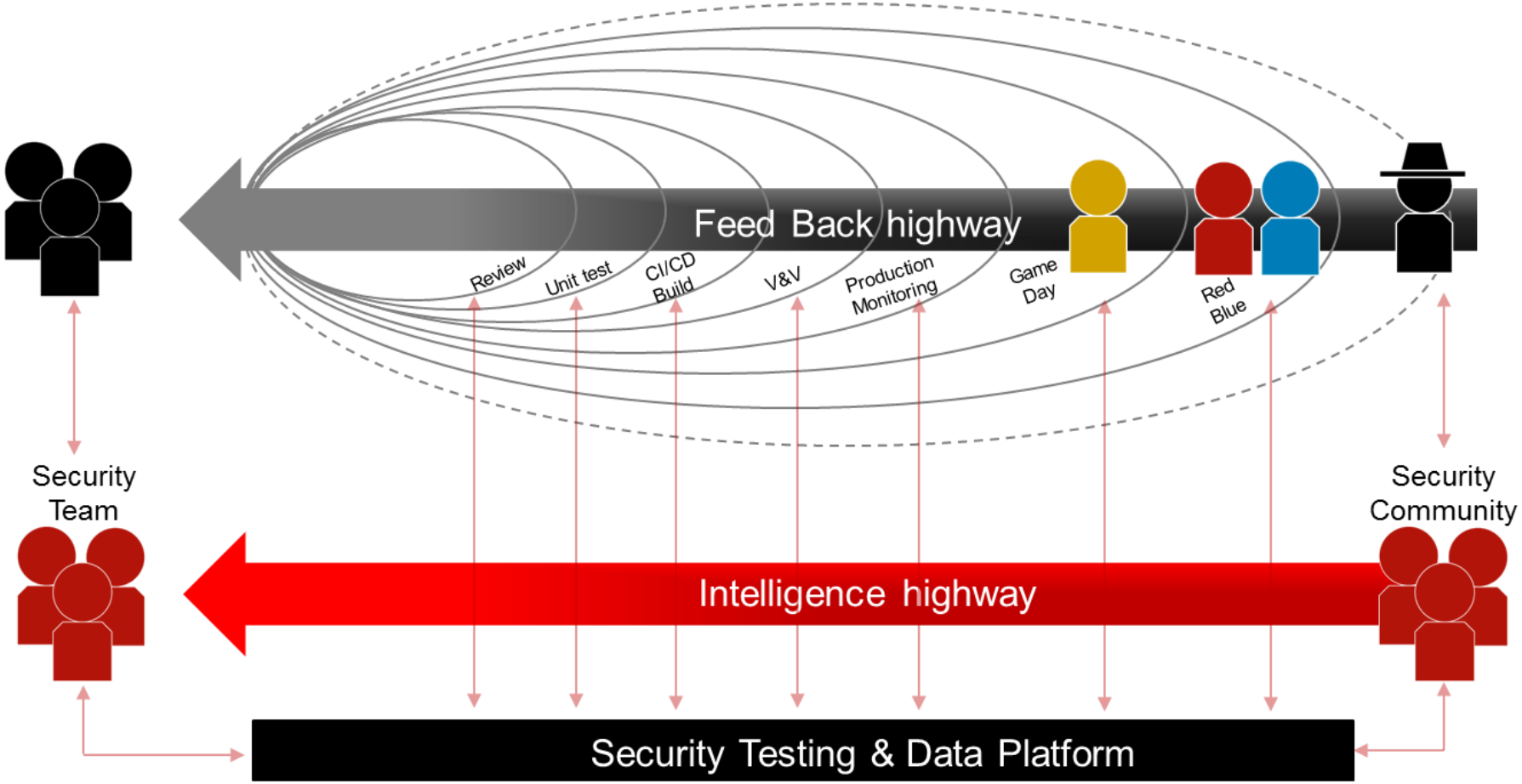
DSB #1: Software Factory

Securing the Factory



DSB #1: Software Factory
Relentless Improvement for Factory

DevSecOps: Seamlessly integrate security into the implementation pipeline; ensuring everyone takes responsibility while continuing to shorten feedback loops



DSB #1: Software Factory
Current State

There is not a standard definition of Software Factory

Common Questions:

- Is there a single factory or multiple?
- Is the factory Government or Supplier owned?
- How does the factory stay current?
- How do we handle multiple different types of software?
- How do we secure the factory?
- How do we enable reuse in a multi-vendor award or across Supplier and Government?

Current incentive structure does not support software factory approach

Current funding approach funds projects, not value streams

Metrics support inputs, not business outcomes

Attack surface is growing exponentially

Technology change is exponential while organizational change is logarithmic

DSB #1: Software Factory Obstacles 1

Recommendation Area	DoD Obstacles	Industry Obstacles
Intellectual property	Lack of agreement on data rights that satisfy both Government and Supplier needs	Lack of agreement on data rights that satisfy both Government and Supplier needs
Contract Language • **Contract for outcomes** • **Contract for capacity**	• Access to technical expertise • Interoperability of software factories (Internal and external)	• Detailed Statement of Work (SoW) vs. Statement of Objectives (SoO) which forces building for obsolescence • Functional-based WBS forces handoffs • How to differentiate from other vendors
Fund value streams as opposed to projects	• Completely different from current Procedures • Difficult to get funding without end-to-end project plan and deliverables by vendor	• Obtaining fair share of value stream funding • Difficult to forecast work because it would not be discreet project with beginning and end
Continuous working and executing software factory	• Difficult to validate • Interoperability • Rapidly changing technology • Approval to Operate (ATO) for different programs • There is not a standardized approach	• Not incentivized to invest in software factory infrastructure. • Customized RFP section L&M from different organizations can lead to a lot of complexity • Not all software is the same

DSB #1: Software Factory Obstacles 2

Recommendation Area	DoD Obstacles	Industry Obstacles
Publish blueprints and playbooks	Require Government resources in a resource constrained environment to govern and evaluate	Difficult to differentiate from other Suppliers
Transparent Integrated PMB	• Limited number of examples on Government programs • Standards not defined, so there is variability across vendors	• Availability of trained PMs with experience in software factories. • Trained Program Management Office (PMO), Contracts, Subcontracts, Finance, Scheduling
Securing the factory	• New vulnerabilities identified daily • Standardized factory provides threat actor time to breach the security • Difficult to validate pedigree of all FOSS/COTS/GOTS/Supplier components	• New vulnerabilities identified daily • Standardized factory provides threat actor time to breach the security • Difficult to validate pedigree of all FOSS/COTS/GOTS/Supplier components
Measure results against practices and processes	• Requires resources to perform analysis in a resource-constrained environment • Access to Subject Matter Experts (SMEs) to validate results	• Multiple different environments, results of practices will vary based on category of what is being built. • Access to SME's

DSB #1: Software Factory Obstacles 3

Recommendation Area	DoD Obstacles	Industry Obstacles
Government run retrospectives	• Requires resources and expertise to perform analysis in a resource-constrained environment	• Potential risk based upon results • Difficult to differentiate amongst vendors
Teams as a service (CID Cells)	• Security • Potential increase in travel budgets • Capability to contract	• Security • Workforce planning
Interim Approval to Operate (IATO) for infrastructure	• Multiple types of software being built • Rapidly changing technology • Multiple different domains in DoD with localized approach to IATO	• Rapidly changing technology; how to keep current tools approved

DSB #1: Software Factory Path Forward 1

Initiative	Action Plan
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)

DSB #1: Software Factory Path Forward 2

Initiative	Action Plan
Publish blueprints and playbooks	• Collaborate with Industry to obtain software factory blueprints and playbooks and publish for use across programs to increase success
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 Path Forward 3

Initiative	Action Plan
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 ○ 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

Continuous Iterative Development

… identify Minimum Viable Product (MVP) approaches … to:

- deliver a series of viable products (starting with MVP) followed by successive Next Viable Products (NVPs);
- establish MVP and the equivalent of a product manager for each program in its formal acquisition strategy, and arrange for the warfighter to adopt the Initial Operational Capability (IOC) as an MVP for evaluation and feedback; and
- engage Congress to change statutes to transition Configuration Steering Boards (CSB) to support rapid iterative approaches

… require all programs entering Milestone B to implement these iterative processes for Acquisition Category (ACAT) I, II, and III programs.

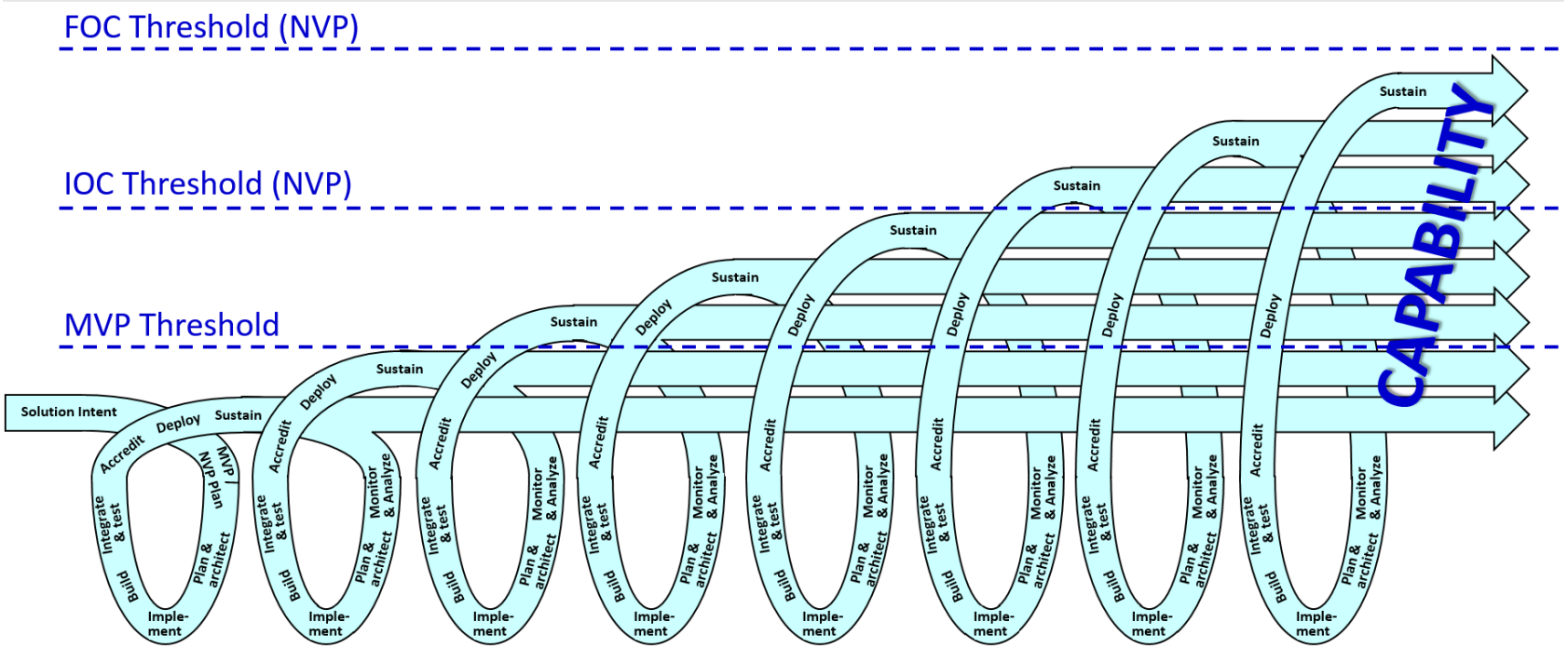
DSB #2: Continuous Iterative Development Assumptions

- **DoD will lead effort to define an IP approach that meets both Government and Supplier needs**
- **Software factories include people, processes and tools, not just a tool chain**
- **Suppliers will have internal factories that match a software factory definition approved by the Government and matching factories inside of agencies.**
- **The CID approach will move away from a winner take all approach and facilitate the inclusion of capabilities from multiple Suppliers in a single system**

DSB #2: Continuous Iterative Development Picture of Success (end-state) 1

Government/Supplier Interface:

- New programs defined by solution intent
- Contracts defined by MVP and NVP plans
- Funding supports capability evolution
- Stakeholders actively engaged in CID lifecycle
- Design guided by Modular Open Systems Approach (MOSA)
- Government access to source code with negotiated IP protections



Program Execution:

- Multi-discipline CID execution includes milestones
- Direct user interaction informs design
- Test automation accelerates delivery

DSB #2: Continuous Iterative Development Picture of Success (end-state) 2

Government/Supplier Interface

- **New programs defined by solution intent:** New programs are defined via a DoDAF CV-1/Solution Intent (Vision, Constraints and Guidance, Goals and Capabilities) and a CV-1/Roadmap showing the general order of desired capability needs. Additional decomposition and elaboration of the need is developed iteratively between Government and Suppliers through the down-select process, with an eye to maintaining a robust set based design space. CSBs operate at the level of the CV-1/Solution Intent and (perhaps) major delivery milestones on the roadmap. Steering at lower levels is integrated with roadmap updates and MVP/NVP planning.
- **Contracts defined by MVP:** Contracting approach includes mechanisms for flexibly defining and approving MVP/NVP capabilities. Contracting / Planning approach acknowledges the need for and enables the development of frameworks for supporting capability sets.
- **Funding supports capability evolution:** Funding structure allows for development of new capabilities and sustainment and evolution of previously delivered capabilities to be done seamlessly, by the same teams and on the same contract.

DSB #2: Continuous Iterative Development Picture of Success (end-state) 3

Government/Supplier Interface (cont'd)

- **Stakeholders actively engaged in CID lifecycle:** Program management norms include the role of an Government Product Manager/Chief Product Owner with defined roles and responsibilities, and clear relations to the Government PM and end users who provide input on design, as well as counterparts (PM Proxy, Chief Engineer, etc.) on the Supplier side. Certification authorities are on board with CID concepts and certification documentation and procedures are appropriate for a CID lifecycle.
- **Design guided by MOSA:** Constraints and guidance include the integration of MOSA tenets (modularity, well defined interfaces) that enable flexibility and reduce risk.
- **Government access to source code with negotiated IP protections:** Contracting approaches protect Supplier IP while providing the Government access to source code for analysis and evolution.

DSB #2: Continuous Iterative Development Picture of Success (end-state) 4

Program Execution

- **Multidiscipline CID execution includes milestones:** Multi-discipline CID teams operate across all aspects of the program including technical milestones. Contract Data Requirement List (CDRL) items and events are aligned with and defined appropriately for the CID lifecycle
- **Direct user interaction informs design:** Suppliers work directly with users and warfighters, delivering systems iteratively in order to meet minimum needs now, and build on small successes with increasingly complex systems. Government Chief Product Owner adjudicates conflicting end user needs and priorities. Government has resources and procedures in place to allow controlled exercise of and rapid feedback on new capabilities.
- **Test automation accelerates delivery:** Test automation at the system level, allow for new system changes to be automatically verified/validated, so that they may be rapidly released or deployed to users (i.e. Continuous delivery / deployment) without delay.
- **Accelerated capabilities for certification and accreditation:** Authorities in place and able to process product releases (NVPs) at the speed of relevance.

DSB #2: Continuous Iterative Development Picture of Success (end-state) 5

Program Execution (cont'd)

- The "idealized" future-state adopts the following sequencing of software products/systems :
  1) Perform up-front systems engineering activities, including a refactored System Requirements Review (SRR) and Preliminary Design Review (PDR), focused on an initial "increment 0" architecture – a hi-level ('1 inch deep') architecture that specifies the major elements/subsystems/interfaces (think MOSA). This includes test architecture, etc.
  2) In parallel with 1), instantiate the infrastructure, tooling and environments, to support CID … i.e. stand-up the software factory
  3) After PDR, apply CID practices (agile workflow e.g. Scrum, test driven development, continuous delivery, continuous accreditation, etc.) to implement the MVP (i.e., the minimum system capability that provides operational utility). The MVP/NVP may or may not be deployed to operations; it is a contracting construct that formally defines the initial development completion criteria. Government oversight will be achieved via incremental reviews and demonstrations integrated with the CID operational cadence.
  4) Continue with 3), adding Government-prioritized features, fixes, security updates, etc. This never ends, as "Software is immortal," at least until the software/system is decommissioned.

DSB #2: Continuous Iterative Development Current State 1

New programs are usually defined as a set of enumerated requirements. Use of high-level constructs (SoO, Capabilities Description Document (CDD), etc.) are sporadic

Program definitions are for the entire program life (~5 years or more) at the same level of detail for the entire program. Some agencies have begun to employ single award Indefinite Delivery/Indefinite Quantity (IDIQ) contracts with semi-annual or annual Task Orders as an alternative

Program awards are usually winner-take-all, fly-offs are uncommon. Set based design and MOSA tenets of open design are not baked into acquisition

In general, contracting mechanisms do not support flexibility in implementing mission needs. Sell-off is to enumerated requirements, and is typically done at the end of the program. Some agencies/programs have begun buying capacity as an alternative.

MVP/NVP planning is not done as part of acquisition process

Changes at low levels (specific requirements, delivery dates for specific functions etc.) require contractual modifications

DSB #2: Continuous Iterative Development Current State 2

Most customers do not have the capability to accept frequent deliveries – no standing operational test environment, no rapid ATO approach

Certification authorities expect traditional artifacts and assume traditional timelines

Systems are delivered to DoD in "big batch releases"

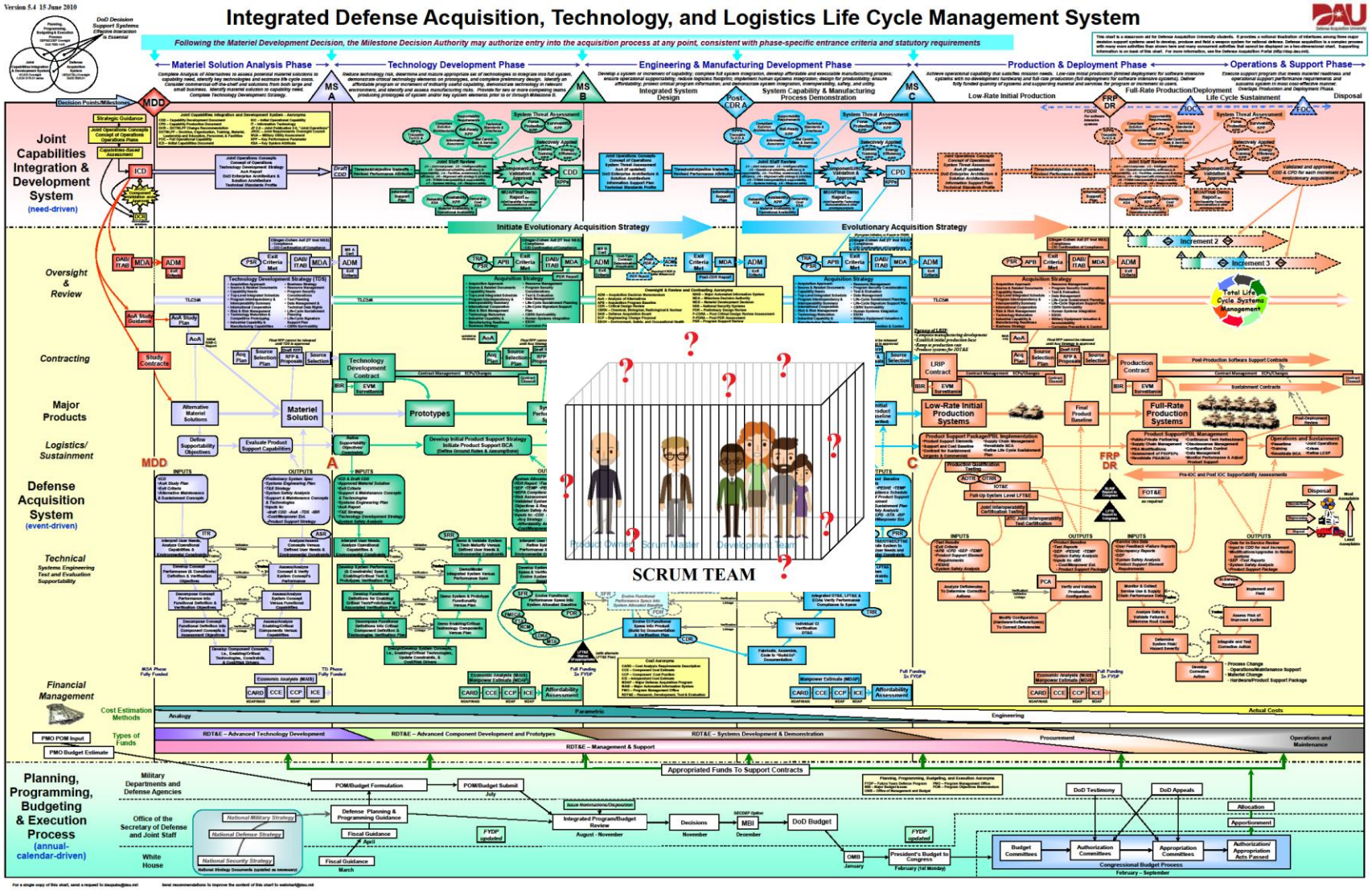
Scope changes require contract modifications

Systems Engineering Technical Review (SETR) events drive 'waterfall' behaviors

Lengthy/tedious ATO process does not support frequent delivery

Systems engineering, infrastructure design, integration and other activities necessary to the delivery of system capabilities are silo-ed and separately managed based on Government requirements.

DSB #2: Continuous Iterative Development Current State 3



DSB #2: Continuous Iterative Development Obstacles 1

Recommendation Area	DoD Obstacles	Industry Obstacles
Lack of trained and experienced personnel on CID (government and supplier)	• Lack of personnel available to be trained and perform tasks • Lack of clarity on planned approach and required training	Lack of clarity on planned approach and required training
Program definitions do not support CID	Current DoD program definition approaches are not CID-friendly (e.g. enumerated requirements, sell off at end)	Current SoWs, metrics and Contracting Officer's Technical Representative (COTR) expectations drive Suppliers towards waterfall approaches
Contracting mechanisms do not support CID	Current common DoD contract types are not explicitly designed for CID operation. *TechFAR Handbook* and Digital Playbook instructions do not appear to be exerting influence on program operations.	Current DoD contracts typically require completion sell off to requirements at the end of the program and do not support changing capability needs/priorities

DSB #2: Continuous Iterative Development Obstacles 2

Recommendation Area	DoD Obstacles	Industry Obstacles
Sell off mechanisms do not support CID	Incremental sell-off not clearly defined or widely used. Acceptance Test Driven Development (ATDD) approaches not integrated into technical management approaches	ATDD approaches not customized for large, complex programs and into technical management paradigm
Government does not have facilities and processes for accepting / testing frequent NVPs	• DoD has not invested in operationally relevant environments into which MVP/NVP deliveries can be made • Whether continuous delivery to actual operations environments is desirable is undecided • Certification, accreditation and acceptance processes do not support frequent delivery	
IP and data rights concerns	Lack of agreement on data rights that satisfy both Government and Supplier needs	Lack of agreement on data rights that satisfy both Government and Supplier needs

DSB #2: Continuous Iterative Development Obstacles 3

Recommendation Area	DoD Obstacles	Industry Obstacles
Certification authorities expect traditional artifacts and assume traditional timelines	Existing certification / authorization organization, process and timeline do not support CID	Suppliers cannot schedule and achieve certification / authorization in line with a CID cadence
No incentive for Suppliers to embrace modularity, openness and set based design practices that enable CID and also lead to a more level playing field and less vendor-lock	Current cost profiles and expectations do not account for the cost frontloading required for CID	Incentives not written into contracts
Current funding approaches do not allow a software factory or CID pipeline to operate at optimal capacity		Changing funding profiles by contract and the inability to share resources between contracts efficiently limit software factory efficiency

DSB #2: Continuous Iterative Development Path Forward

Initiative	Action Plan
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
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
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

DSB #3a: Risk Reduction

Risk Reduction and Metrics for New Programs

… allow **multiple vendors** to begin work. A down-select should happen after at least one vendor has proven they can do the work, and should retain several vendors through development to reduce risk, as feasible.

… **modernize** cost and schedule estimates and measurements. They should evolve from a pure Source Lines of Code (SLOC) approach to historical comparables as a measurement, and should adopt the National Reconnaissance Office (NRO) approach of contracting with the defense industrial base for work breakdown schedule data to include, among others, staff, cost, and productivity.

… require the PM to build a **framework** for status estimation. **Example metrics** include:

- Sprint burndown
- Epic and release burndown
- Velocity
- Control chart
- Cumulative flow diagram

DSB #3a: Risk Reduction Assumptions

All members of the Supplier/Government team have training and a common understanding of software CID and a software factories

An enterprise open architecture and business case can be made that enables competition on critical and high cost components

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

Funding and contracts can be aligned in a timely manner to support implementation and/or migration to software factories

Factory boundaries will be able to handle all software and hardware elements of the full legacy system

DSB #3a: Risk Reduction
Picture of Success (end-state)

Start the program right by creating an affordable win-win partnership through the acquisition and support strategy with common goals and objectives that enable the team to proceed at acceptable risk

- Have defined activities, checks and balances, and demonstrations throughout the lifecycle for the Government to provide feedback on developed products
- Have objective, quantitative "definitions of done" and evaluation criteria
- Maintain an open architecture and competition on critical components
- Have evaluation criteria for continuing with multiple Suppliers and well defined exit criteria for down-selects

Develop and track metrics to effectively control processes, measure against goals, objectives, risks, and make decisions

- Enable continuous improvement

An IP agreement for the project that meets both Government and Industry needs throughout the system lifecycle is in place

DSB #3a: Risk Reduction
Current State

Competitive prototyping (CP): a good approach to rapidly assess potential solutions and reduce risks, but it generates other risks and/or issues that must be addressed in program planning

- CP is more expensive for the Government in the short run
  - Requires more up-front investments due to multiple Suppliers
  - Requires greater resources (funding, staffing, tools, environments)
  - Total lifecycle costs may be reduced on the back end of the lifecycle, but the Government doesn't have sufficient data to demonstrate this.
- Not all technical problems may be solved by down-select – some problems persist
  - The Government often doesn't have the technical expertise and competency to distinguish between "good" and "bad" (including design).
  - Competitive prototyping is unlikely to address non-functional requirements, including quality attributes like maintainability, reliability, and security.
  - Competitive prototyping should resolve key risks and demonstrate key technologies, but is not expected to resolve all risks nor demonstrate operational capabilities.

DSB #3a: Risk Reduction
Competitive Prototyping Survey, 2008 (USC CSSE)

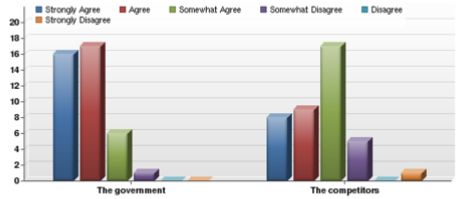
Microsoft Point 97-2003 Presei

Study indicates that CP can help in many situations, but has a number of pitfalls. CP does not solve all acquisition problems.

Competitive Prototyping Survey Results Overview (Draft)

Dan Ingold
Center for Systems and Software Engineering
University of Southern California

CP efforts are on balance good for

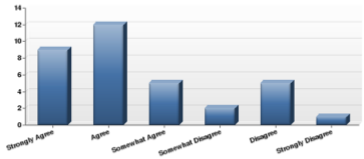


- General agreement that CP is good for both government and industry
- Responses indicate somewhat better deal for government
- No consensus from interviews on why the perceived difference

DoD does/should use CP (cont'd)…

- Primary reason is to reduce risk
- Risk reduction drives cost/schedule improvement
  - In the long-run
  - Short-run cost/schedule is higher and longer
- May be other benefits, like improved trust
  - An implicit benefit, not exactly briefed up the line
- Unstated benefit is to encourage innovation
- Not a panacea, just good management practice

CP development should be evolutionary or iterative, with ability to provide interim operational capabilities

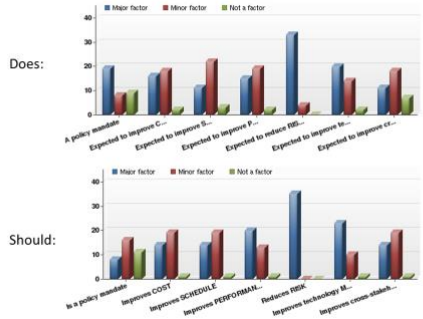


- Mixed response, but tending toward agreement
- Response perhaps due to use of "operational capabilities" phrase, which several interviewees took issue with
- Interviewees who disagreed would have agreed if phrase omitted

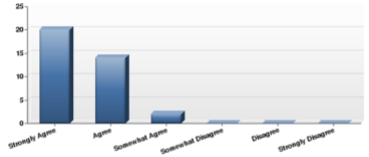
Common themes among responses

- Many interviewees stressed the importance of *building relationships* with the customer
- *Ongoing collaboration* between stakeholders and contractors is vital for successful outcomes
- Most cited need for flexibility, and use of *agile or evolutionary approaches*
- There is no substitute for *technical expertise* within (not subcontracted to) the PMO, which is now lacking
- Something must be done to *keep teams funded* during the down-select intervals

DoD does/should use CP because it is…

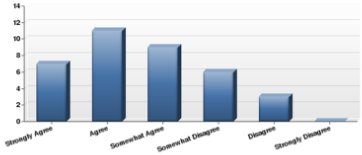


CP should encourage active participation of stakeholders



- Among strongest agreement in entire survey
- Interviewees said end-user/stakeholder participation was vital
- Inconsistent with "level playing field" concern in earlier question?
- Interviewee: "Access to stakeholders should be fair and equal, but competitors should not be all in one room asking questions"

CP will require significantly larger investments in evaluation tools, test beds, and capabilities within acquiring organizations



- Broad response, but tending toward agreement
- Several interviewees thought tools and test beds should be at competitors, not acquiring organizations, possibly through FRDC or IV&V
- Tools and test beds at competitors requires transparency to ensure complete, correct and fair evaluation
- One interviewee observed FRDC/IV&V organizations lack "skin in the game", making them less effective evaluators

DSB #3a: Risk Reduction Obstacles 1

Recommendation Area	DoD Obstacles	Industry Obstacles

CP strategy	• Lack of early systems engineering and technical expertise
	• Optimistic framing assumptions on reduction of risk from prototyping
	• Subjective and qualitative linkage to RFP (RFP sec. L&M)
	• Acquisition communications across competing teams during program execution, consistent messaging
	• Clarity of how risk reduction/down-selects vary for traditional vs. software CID
		• Lack of agreement on data rights that satisfy both Government and Supplier needs
		• Competing priorities for investment & resources with uncertain outcomes (risk vs. reward)
		• Bad business case
		  • Low probability of the competition occurring and/or low probability of winning
		  • Low margins and/or revenues projections

DSB #3a: Risk Reduction Obstacles 2

Recommendation Area	DoD Obstacles	Industry Obstacles
Risk and opportunity planning and execution	• Lack consistent objective measures to measure acquisition risk and drive risk reduction decisions • Alignment of contract incentives and measures • Integrating risk and opportunity into plans	• Alignment of contract incentives and measures • Competitive pressures • Integrating risk and opportunity into plans
Resources	• Resources for acquiring, funding, and managing multiple procurements concurrently • Depth and breadth of qualified personnel • Keeping competing teams funded during down select decisions	• Applying best resources (staff, funding, Internal Research and Development (IR&D), tools, environments) to pursuits with best win probability (p_{win}) and Return on Investment (ROI) • Competing priorities for resources • Conflicts with company strategic plans • Keeping teams intact and funded during down select decisions

DSB #3a: Risk Reduction Obstacles 3

Recommendation Area	DoD Obstacles	Industry Obstacles
Stakeholder engagement	• Creating the foundation for trusting relationship • Understanding stakeholders interest, expectation, and requirements • Building and improving a trusted relationship • Balancing and integrating stakeholder needs • Developing partnerships	• Creating the foundation for trusting relationship • Understanding stakeholders interest, expectation, and requirements • Building and improving a trusted relationship • Balancing and integrating stakeholder needs • Developing partnerships

DSB #3a: Risk Reduction Path Forward 1

Initiative	Action Plan
Acquisition strategy	Acquisition strategies that provide a fair opportunity to compete, retain competition throughout the lifecycle for critical components to enable rapid evolution of the product.
Competitive prototyping	• Review analyses/reports from prior DoD competitive prototyping initiatives, and integrate lessons learned into action plan for DSB recommendations. • Competitive prototyping risk reduction strategy should account for both functional and non-functional requirements.
Cultural shift	Migrate from subjective qualitative assessment to objective quantitative assessment of risk that support business decisions
Resources	DoD investment to acquire, deploy, integrate, and maintain evaluation tools and test beds
Workforce development	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.

DSB #3a: Risk Reduction Path Forward 2

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 #3b: Metrics

Risk Reduction and Metrics for New Programs

… allow **multiple vendors** to begin work. A down-select should happen after at least one vendor has proven they can do the work, and should retain several vendors through development to reduce risk, as feasible.

… **modernize** cost and schedule estimates and measurements. They should evolve from a pure SLOC approach to historical comparables as a measurement, and should adopt the NRO approach of contracting with the defense industrial base for work breakdown schedule data to include, among others, staff, cost, and productivity.

… require the PM to build a **framework** for status estimation. **Example metrics** include:

- Sprint burndown
- Epic and release burndown
- Velocity
- Control chart
- Cumulative flow diagram

DSB #3b: Metrics Assumptions

Traditional software measures based on waterfall development are generally not well suited for adapting to software CID.

Some historical measures (e.g., SLOC) may still be necessary for collection and analysis across programs while the defense industry transitions.

Measures for software CID are common in industry, but there is not a universal, consensus, or standard set of measures implemented across companies or defense programs.

Measurement frameworks should be based on information needs adapted to the characteristics of the program or acquisition

- But it is reasonable to also offer a palette of appropriate candidate measures to choose from based on use and effectiveness in industry

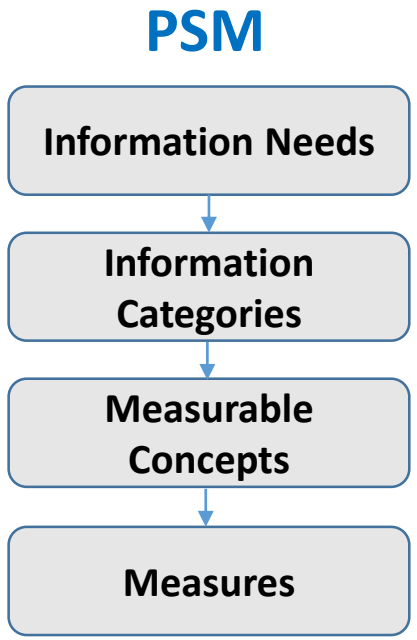
DSB #3b: Metrics
Picture of Success (end-state) 1

Consensus Government/Industry measurement framework and common measures applied across defense software acquisition programs.

- Measures aligned top-down from business objectives, information needs, performance targets, quality attributes
- Measures used for objective oversight and management decision-making

Transition to modernized software measures aligned with current practices for CID.

- Derived from and consistent with software factory processes
- Migration away from traditional legacy measures (e.g., phases, incentives based on documentation and milestones)



DSB #3b: Metrics
Picture of Success (end-state) 2

Software program objectives and measures aligned and tailored from common frameworks for aggregation across programs.

Historical repositories of consistent software performance measures maintained within defense industry companies and a subset collected in DoD repositories to support basis of estimates, proposals, and program monitoring.

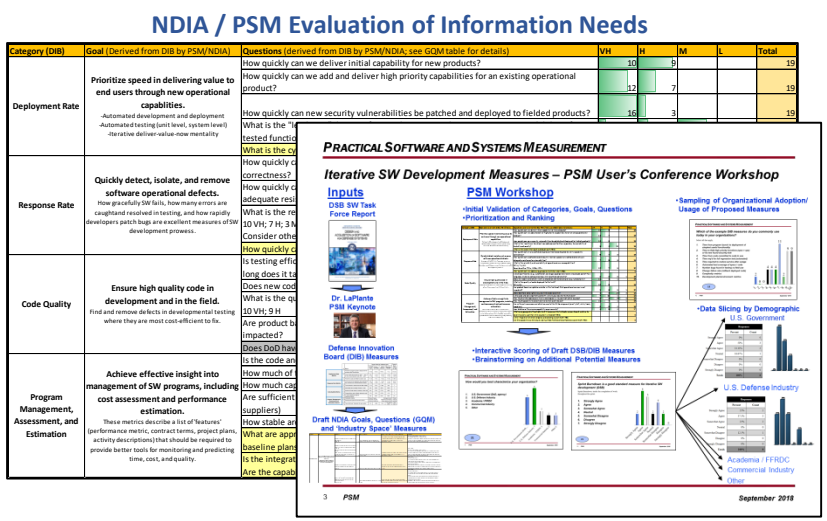
DSB #3b: Metrics Current State

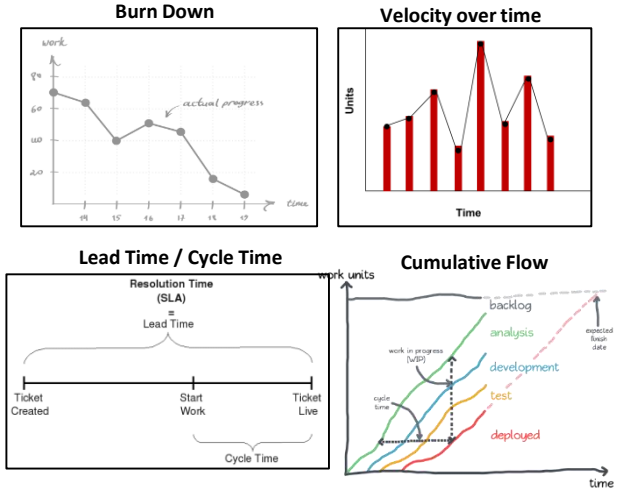
Implementation of CID measures is inconsistent within companies and across defense programs.

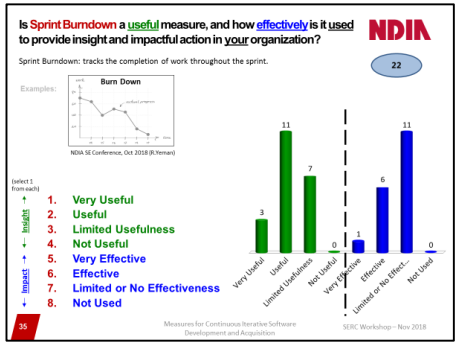
Transition to DevSecOps is in progress and widespread, but few measure-ment standards or operational definitions are in broad use yet across the defense industry. Many are program-specific

Initiating draft framework of information needs and measures for CID, derived from DSB and Defense Innovation Board (DIB) recommendations and industry practice.

Measures Evaluated	
DSB	**DIB**
Sprint burndown	Time from program launch to deployment of simplest useful functionality
Epic and Release Burndown	Time to field high priority fcn (spec > ops)
Velocity	Time to fix newly found security hold (find >ops)
Cycle Time (Control Chart)	Time from code committed to code in use
Cumulative Flow Diagram	Time req'd for full regression test (automated) and cyber audit/ penetration testing
	Time required to restore service after outage
	Automated test coverage of specs/code
	Number of bugs caught in testing vs. field use
	Change failure rate (rollback deployed code)
	Complexity metrics
	Development plan/environment metrics







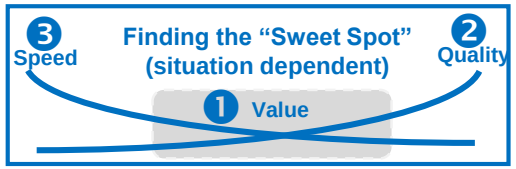
Interactive Workshops and Surveys of Agile Measurement Practices (NDIA, PSM, INCOSE, SERC)

- **Information needs / objectives**
- **Measures (usefulness, effectiveness)**

DSB #3b: Metrics

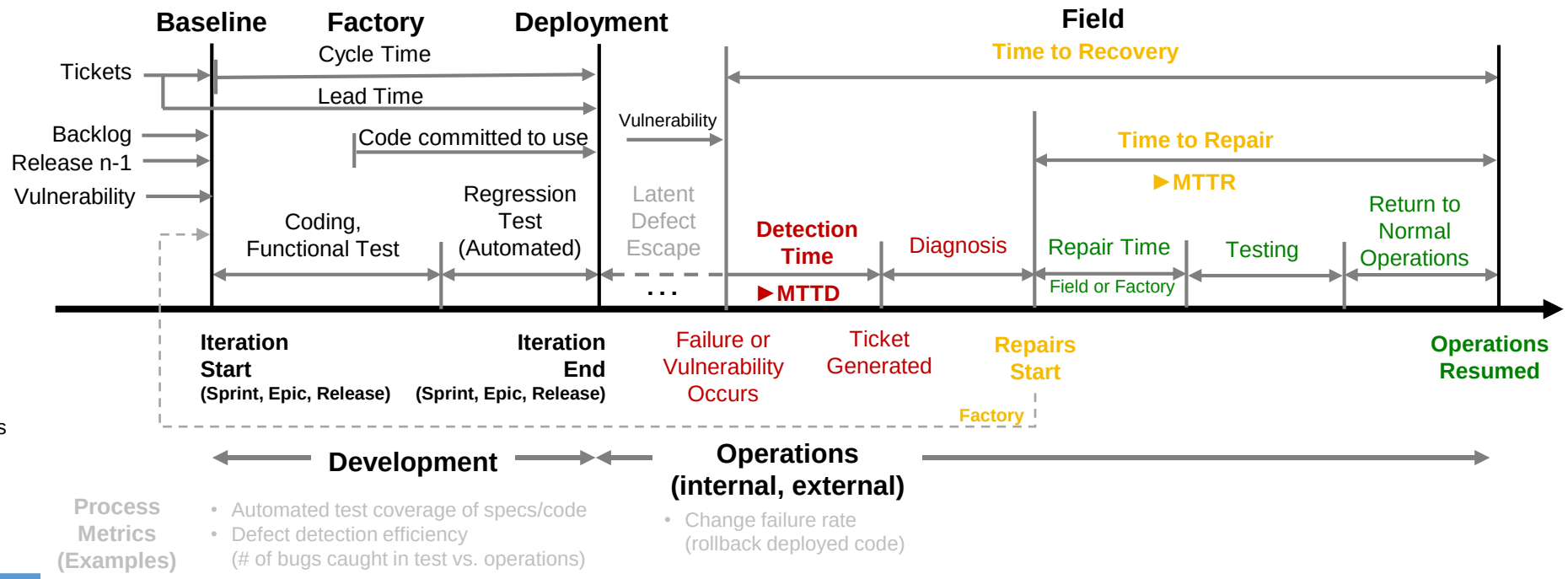
Frameworks for aligning measures with objectives

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



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

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



Summary of DIB Metrics Categories

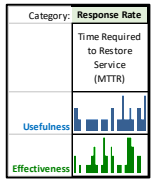
Deployment Rate
- Initial launch to deployment of simplest useful functionality *[MVP]*
- Time to field high priority fn (spec>ops) or security hole (find>ops)
- Time from code committed to code in use

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

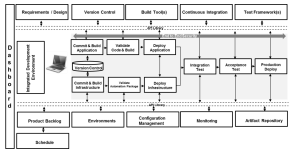
Code Quality
- Automated test coverage of specs / code
- Number of bugs caught in testing vs. field use *[defect detection %]*
- Change failure rate (rollback)

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

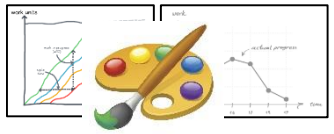
Industry feedback (usefulness, effectiveness)



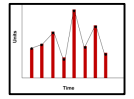
Available measures instrumented and automated by the toolchain



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



Enterprise measures driven by business performance objectives



productivity, quality, estimate accuracy, …

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

Adoption

References:
- Defense Science Board, Design and Acquisition of Software for Defense Systems, Feb 2018
- Defense Innovation Board Metrics for Software Development, version 0.9, 9 Jul 2018
- MTTR, MTBF, or MTTF? A Simple Guide to Failure Metrics. https://limblecmms.com/blog/mttr-mtbf-mttf-guide-to-failure-metrics/

DSB #3b: Metrics
EVM for CID Programs

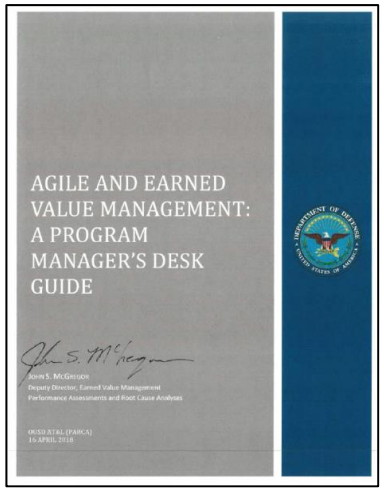
Earned Value Management (EVM) has often problematic when applied to agile or software CID programs, with uncertain delivery outcomes that may vary based on iteration, priorities, and stakeholder input.

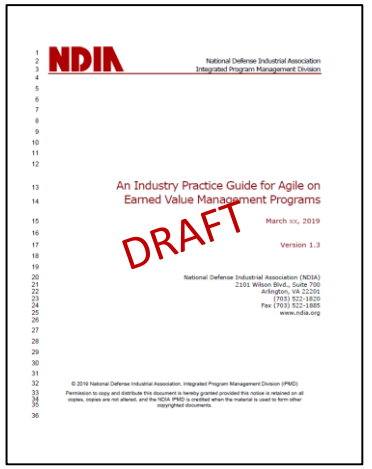
Where EVM is required, NDIA recommends aligning EVM with emerging best practices: (PARCA, NDIA, SEI, …)

- Contracting for CID programs
- Integration of CID planning with EVM processes and PMB (e.g., WBS, IMP, IMS, ETC/EAC, BCWP, BCWS, ACWP, % complete tracking, CV, SV)
- Adapt CID measures for planning and managing work package EVM performance
- Using EVM to managing baseline changes on CID programs

Recommend tailoring DoD policies and guidance (e.g., DODI 5000.02) to better integrate effective PM practices for software-intensive programs.

Until DoD and industry can develop better tools (e.g., section 809), EVM still applies for managing CID vs. a performance baseline





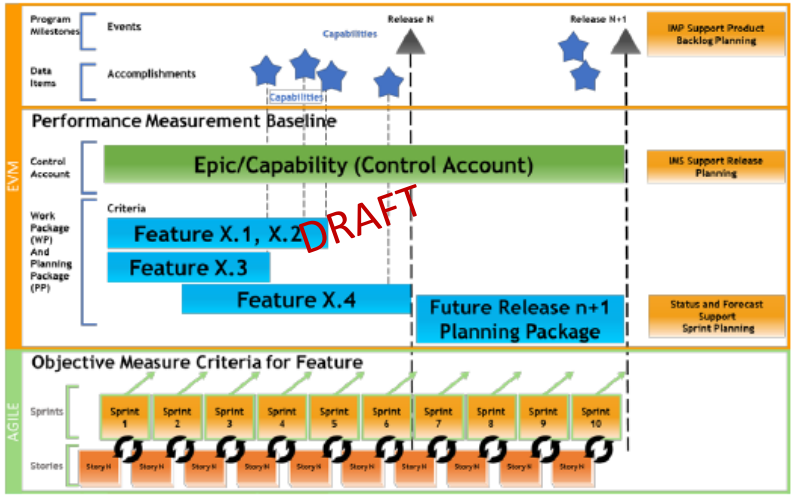


Figure 2-1: Agile IMP Event to EVMS Hierarchies. In this example, IMP events are equivalent to Customer Releases, with Significant Accomplishments and Accomplishment Criteria representing delivered capabilities delivered in Work Packages where Features are implemented.

References:

1. *An Industry Practice guide for Agile on Earned Value Management Programs.* NDIA Integrated Program Management Division. March 2019.
2. *Agile and Earned Value Management: A Program Manager's Desk Guide*. OUASD AT&L (PARCA). April 2018.
3. *RFP Patterns and Techniques for Successful Agile Contracting.* Software Engineering Institute, CMU/SEI-2016-SR-02.

DSB #3b: Metrics Obstacles 1

Recommendation Area	DoD Obstacles	Industry Obstacles
Modernizing software practices and measures for CID	• Long learning curve - workforce inexperience • DIB: "*Most SW teams are attempting to implement DevOps and 'agile' approaches but in most cases the capabilities are still nascent (and hence fragile)*"	• Operations is not often under industry control – industry focus may be more focused on factory and pre-deployment • Long learning curve - workforce inexperience • Pockets of experience > enterprise transition • Leveraging early successes and knowledge across diverse programs and domains
Measurement framework for CID	• Defining clear operational definitions • Contracting language, adoption in acquisition	Some CID /DevSecOps measures are used in some programs and companies, but far from a consistent or consensus approach
Repository of historical measures	IP, proprietary data	Current historical databases are largely for traditional development - lack robust historical DBs for CID, DevSecOps

DSB #3b: Metrics Obstacles 2

Recommendation Area	DoD Obstacles	Industry Obstacles
Competitive prototyping	Resources for acquiring, funding, and managing multiple procurements concurrently	Investment, Research & Development (R&D) needed for effective software factories depends on a reliable business case
Patience with transition timing – long term vision, not a quick fix	• Specifying contracting language to incentivize Suppliers to provide modern measures to program office and DoD repositories • Major investment needed (tools, training, culture, …)	Adoption - expand penetration of DevSecOps measurable performance across programs by building on initial successes Mutual benefits of data sharing with DoD (ROI)
Cultural obstacles	DIB Do's and Don'ts for Software https://media.defense.gov/2018/Nov/02/2002058905/-1/-1/0/DIB_DOS_DONTS_%20SOFTWARE_V2_2018.11.02.PDF	**Culture** In this final section we catalog a list of culture items that do not necessarily require changes in statutes, regulations, or instructions, but rather a change in the way that DoD personnel interpret implement their processes. Changing the culture of DoD is a complex process, depending in large part on incentivizing the behaviors that will lead to the desired state. Data and metrics • Multiple, competing, and sometimes conflicting types of data and metrics used, or not used, for assessing software in DOD • Inability to collect meaningful data about software development and performance in a low cost manner, at scale • Inability to turn data into meaningful analysis and inability to implement decisions or changes to software activities (L/R/C)

DSB #3b: Metrics Path Forward

Initiative	Action Plan
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: Program Transition

Current and Legacy Programs in Development, Production, and Sustainment

For ongoing development programs, … plan transition to a software factory and CID.

For legacy programs where development is complete, … make the business case for whether to transition the program.

DSB #4: Program Transition
Picture of Success (end-state) 1

Type	Success	Criteria	Verification Approach
People	**Skill assessment**	Skill assessment strategy across key areas to assess current skills and path to acquire additional skills. People can level up skills or experience.	• Review Skill assessment matrix with path to level-up • Previous results of past transitioned programs.
People	**Educated workforce able to successfully transition**	Legacy programs who are currently in development, production, and sustainment have skills to work in software factory.	• Key Skill List • Blueprint to perform Gap analysis
Process	**Business case and transition plan based on phase of delivery**	Business case criteria for program selection. Playbooks for transition are available based on what phase of delivery and type of product being built.	• Review of business case • Metrics of transitioning program • Success rates of past transitions
Process	**Risk Adjusted Product backlog**	Based on the "AS IS" state of any legacy program a risk profile and a risk adjusted product backlog are developed to transition programs to software factory with lowest risk.	Review process for developing risk profile and risk-adjusted product backlog.

DSB #4: Program Transition
Picture of Success (end-state) 2

Type	Success	Criteria	Verification Approach
Process	**Strategies for incrementally building up test automation with minimal disruption.**	Ability to visualize legacy codebase, determine current levels of automation, and suggest a strategy for appropriate level of test automation build-up.	• Static analysis metrics of codebase available. • Metrics available for past transitions
Process	**Assessment of supply chain and Pedigree of software.**	Assessment of all (FOSS/COTS/ GOTS/Supplier) code available to verify pedigree and security.	Strategy and tools available to assess program supply chain
Process	**Playbook and blue prints to incrementally transition legacy code to software factory**	Assess existing code and evaluate whether technology compatible with transition. Develop business case for transition. Recommendation on whether to retain, refactor, re-host, or sunset..	Previous metrics on similar transitions available

DSB #4: Program Transition
Picture of Success (end-state) 3

Type	Success	Criteria	Verification Approach
Tools	**Tool-set(s) available to generate "As Built" documentation and models of existing legacy code.**	Automated support for generation of "As Built" documentation and models to identify current state of software and recommendation for transition to software factory.	Ability to generate As-Built documentation if needed

DSB #4: Program Transition
Current State

Monolithic Architecture(s)

Lengthy and complex certification and accreditation process

Lengthy cycle time to add functionality or fix defects (Months/years vs. days)

High levels of risk to make change to system

Large amounts of technical debt

Majority of testing is manual - low levels of test automation in place

Tightly coupled systems with multiple dependencies

Proprietary portions of system, where you can only see binaries

Large number of vulnerabilities

Program Management not educated in ways to manage or support software delivery

DSB #4: Program Transition Obstacles 1

Recommendation Area	DoD Obstacles	Industry Obstacles
Current Contract Definition	• Legacy programs have an existing contract that does not provide affordances for migration to a software factory • There is not a common software factory definition(s) that will be utilized across DoD	• There is not a common definition across all of the customers on what a software factory includes. • We may need to update contracts for all legacy programs, which can be time-consuming
Funding	• Transitioning monolithic tightly coupled systems to software factory will be costly • Lack of software factory funding	• Funding does not exist for transitioning legacy projects • Funding for up-skilling staff
Transition Risk	• Life-critical systems have existing Technical Readiness Levels (TRLs); we could risk degrading TRL • Operations disruption risk	• Existing TRLs for life-critical systems could be degraded. • Technology may not be compatible with software factory
Software Factory Skills	A shortage of trained individuals in how to manage and execute projects in software factory.	Re-tooling existing teams on legacy programs will take time

DSB #4: Program Transition Obstacles 2

Recommendation Area	DoD Obstacles	Industry Obstacles
Workforce skills assessment	• Common skillset assessment does not exist • There are multiple types of programs, the list will be extensive	• Common skillset list not available • Need to develop or purchase training for skillsets
Supply Chain Pedigree	Not an easy way to evaluate varied software products, in many cases only have access to binaries on legacy programs	• Not an easy way to evaluate varied software products. • Systems have evolved over many years and records may not be available.

DSB #4: Program Transition Path Forward 1

Initiative	Action Plan
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

DSB #4: Program Transition Path Forward 2

Initiative	Action Plan
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 to migrating program components to the software factory

DSB #5: Workforce

Workforce

… develop workforce competency and a deep familiarity of current software development techniques. [Gov't & Industry]

… develop a training curriculum to create and train this cadre of software-informed PMs, sustainers and software acquisition specialists. [Gov't & Industry]

… Chief Executive Officers (CEOs) of DoD prime contractors should brief the USD(A&S) at least annually to demonstrate progress on adapting modern software practices [Industry]

… establish a special software acquisition workforce fund modeled after the Defense Acquisition Workforce Development Fund (DAWDF), … to hire and train a cadre of modern software acquisition experts across the Services. [Gov't]

… create an iterative development IPT with associated training. The Service Chiefs should delegate the role of Product Manager to these IPTs. [Gov't]

DSB #5: Workforce Assumptions

Resources are available to create and train this cadre of software-informed PMs, sustainers and software acquisition specialists to meet demand

Personnel with the required software intensive program expertise can be hired by industry and the Government to meet demand

There is a business case for both the Government and industry for migrating to software CID and software factories that justifies the investments in training

DSB #5: Workforce
Picture of Success (end-state) 1

... develop a training curriculum to create and train this cadre of software-informed PMs, sustainers and software acquisition specialists.

- A new CID, DevSecOps and Modern software-centric systems training curriculum is established at the Defense Acquisition University (DAU) that cuts across career fields (e.g., PMs; sustainers; acquisition specialists)
- Pilots of a modern training academy for all software-centric systems acquisition professionals, developers, and associated functions that relies heavily on Academia, Industry, Commercial Training Solutions and existing best practices.
- An engagement platform / modern community of practice (such as chatOps), provides DoD's software-centric systems practitioners a platform to connect across services, share their software-centric systems experience and skills, communicate through knowledge/chat channels, gather their pain points, and develop solutions leveraging the entire enterprise

DSB #5: Workforce
Picture of Success (end-state) 2

… USD(A&S) has insights into a growing software industrial base that has (a) a demonstrated software factory and delivery capability, and (b) a growing workforce with modern software practice expertise

- Create a trusted relationship where all parties understand stakeholders interests, expectation, and requirements, balance and integrate stakeholder needs, and develop a last partnership
- Consensus Government/industry measurement framework and common measures applied across defense software acquisition programs

… significant increase in hiring, retaining, and training of a cadre of modern software acquisition experts, including CID-, Tech-, and DevSecOps-coaches across the Services through the use of a dedicated DoD Software Acquisition workforce fund.

- Full alignment with the Defense Industrial Base (DIB) -proposed strategic recruiting campaign for civilians with decentralized recruiting supported by a centralized team to align efforts and share best practices

DSB #5: Workforce
Picture of Success (end-state) 3

… a CID IPT, with modern SW expertise, is established within all PMOs; the IPTs and programs are led by a PM experienced/trained in modern software practices and supported by the coaches discussed above.

- The IPTs are building and leveraging a more modern, connected community to build a culture of agility, innovation and entrepreneurship to overcome challenges with digital product delivery across DoD
- Multi-discipline CID teams operate across all aspects of the program including technical milestones. CDRLs and events are aligned with and defined appropriately for the CID lifecycle.
- Ability to map the workforce in software-intensive-systems engineering and manage resources across service boundaries
- Align training and education to current and future needs and target recruiting and personnel development to meet future needs

DSB #5: Workforce
Picture of Success (end-state) 4

Trained and experienced industry partners and supply chain with a common understanding of the Government-planned approach and performance measures for continuous software development (CID)

DoD software-intensive-systems engineering workforce, fully trained and proficient in modern software development competencies

A recruiting pipeline to obtain, train and retain DoDs best and brightest software-intensive-systems engineers to support future program software development needs

DSB #5: Workforce
Current State 1

The Government does not have modern software development expertise in its program offices and broader functional acquisition workforce

- Early system and software engineering is not available to start program right
- Stakeholders are not actively engaged in software CID throughout the lifecycle
- System functionality is not defined to a level that supports rapid CID and deployment

Software-informed PMs, sustainers and software acquisition specialists are not available to plan and execute CID programs

DSB #5: Workforce
Current State 2

Limited ability to manage, track, acquire, and retain a modern software development work force

- No ability to count the workforce in software-intensive-systems engineering
- Needed software-intensive-systems engineering skills capabilities not identified
- Training and education not aligned to current and future needs
- Inability to target recruiting efforts

DSB #5: Workforce Obstacles 1

Recommendation Area	DoD Obstacles	Industry Obstacles
DoD competency skill gaps in modern software development practices	• Lack of understanding of CID / DevSecOps • Lack of modern tool stacks in the Government	• Lack of common understanding of CID / DevSecOps between Government and Industry • Inconsistent approaches across industry and the supply chain
Management of software-intensive-systems engineering resources	• No ability to count the workforce with SW Engineering skills • No Office of Personnel Management (OPM) Software Occupational Job Series • Lack of workforce mobility	• Retention of skilled work force in competitive environment • Retention of workforce on delayed procurements and down selects • Competing priorities for critical assets • Loss of contracts or failure to win competitions • Conflict of interest constraints

DSB #5: Workforce Obstacles 2

Recommendation Area	DoD Obstacles	Industry Obstacles
Align software-intensive-systems engineering skills to needs: What software-intensive-systems engineering skills are needed? IPT structure lacks skill-mix staffing model	Incomplete workforce skill gap analysis and DoD software PMO staffing models	Lack of understanding of government needs and the Supplier responsibilities, authorities and accountability on each contract
Who to recruit? Measures of recruiting success?	Inability to target recruiting	• Lack of clearly defined program responsibilities, accountability, and authorities for CID and DevSecOps partnership with the customer • Clear understanding of contract requirements and risks

DSB #5: Workforce Obstacles 3

Recommendation Area	DoD Obstacles	Industry Obstacles
Role of the Government and role of the Supplier in CID / DevSecOps	• Lack of common understanding of CID / DevSecOps between Government and Industry • Lack of clearly defined responsibilities, accountability, and authorities	• Lack of common understanding of CID / DevSecOps between Government and Industry • Few DevSecOps measurement standards or operational definitions are in broad use yet across the defense industry. • Lack of clearly defined responsibilities, accountability, and authorities • Contract and incentives misalignment

DSB #5: Workforce Path Forward 1

Initiative	Action Plan
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
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

DSB #5: Workforce Path Forward 2

Initiative	Action Plan

Workforce management	• 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

Software is Immortal – Software Sustainment

…direct that RFPs for acquisition programs entering risk reduction and full development should specify the basic elements of the software framework supporting the software factory … These should then be reflected in the source selection criteria for the RFP.

… availability, cost, compatibility, and licensing restrictions of such framework elements to the U.S. Government and its Suppliers should also be part of the selection criteria for contract award.

… proposers may designate pre-existing components not developed under the proposal but used or delivered as part of the project. However, limitations related to use or access to underlying design information may also be a selection criteria.

…except for such pre-existing components, all documentation, test files, coding, application programming interfaces (APIs), design documents, results of fault, performance tests conducted using the framework, and tools developed during the development, as well as the software factory framework, should be:

- delivered to the U.S. Government at each production milestone; or
- escrowed and delivered at such times specified by the U.S. Government (i.e., end of production, contract reward).

…selection preference should be granted based on the ability of the United States to reconstitute the software framework and rebuild binaries, re-run tests, procedures, and tools against delivered software, and documentation. These requirements should flow down to subcontractors and suppliers

DSB #6: Software Sustainment Assumptions

Workforce initiatives (see DSB Recommendation #5) will ensure that there is an available and supported workforce that is fully-trained and proficient in modern software development competencies

The Government establishes incentives and funding to enable more effective organic software infrastructure for the DoD (e.g., Software factories, Software Integration Laboratories (SILs), Sustainment Centers)

Government and Industry collaborate to develop creative approaches to manage IP throughout the entire product life cycle using a "work shared sustainment" approach. This approach recognizes the evolution from Supplier-intensive sustainment (with Government participation) to Government-intensive sustainment (with Supplier participation)

DSB #6: Software Sustainment
Picture of Success (end state) 1

RFPs for acquisition programs entering risk reduction and full development specify the basic elements of the software framework supporting the software factory

- Framework reflected in the source selection criteria for the RFP
- Availability, cost, compatibility, and licensing restrictions of such framework elements to the U.S. Government and its Suppliers are part of the selection criteria for contract award.

Except for such pre-existing components, all documentation, test files, coding, APIs, design documents, results of fault, performance tests conducted using the framework, and tools developed during the development, as well as the software factory framework, are:

- Delivered to the U.S. Government at each production milestone; or
- Escrowed and delivered at such times specified by the U.S. Government (i.e., end of production, contract reward).

DSB #6: Software Sustainment
Picture of Success (end state) 2

Sustainment organizations have ownership and delivery of the technical data packages, software and environments; and can reconstitute the system software

Selection preference granted based on the ability to deliver evolving, viable software capabilities that are sustainable. The government must then be able to reconstitute the software framework and rebuild binaries, re-run tests, procedures, and tools against delivered software, and documentation. These requirements flow down to subcontractors and suppliers.

For the software domain, planning and budget for refactoring and design of the software on a continuous basis are part of the cost baseline.

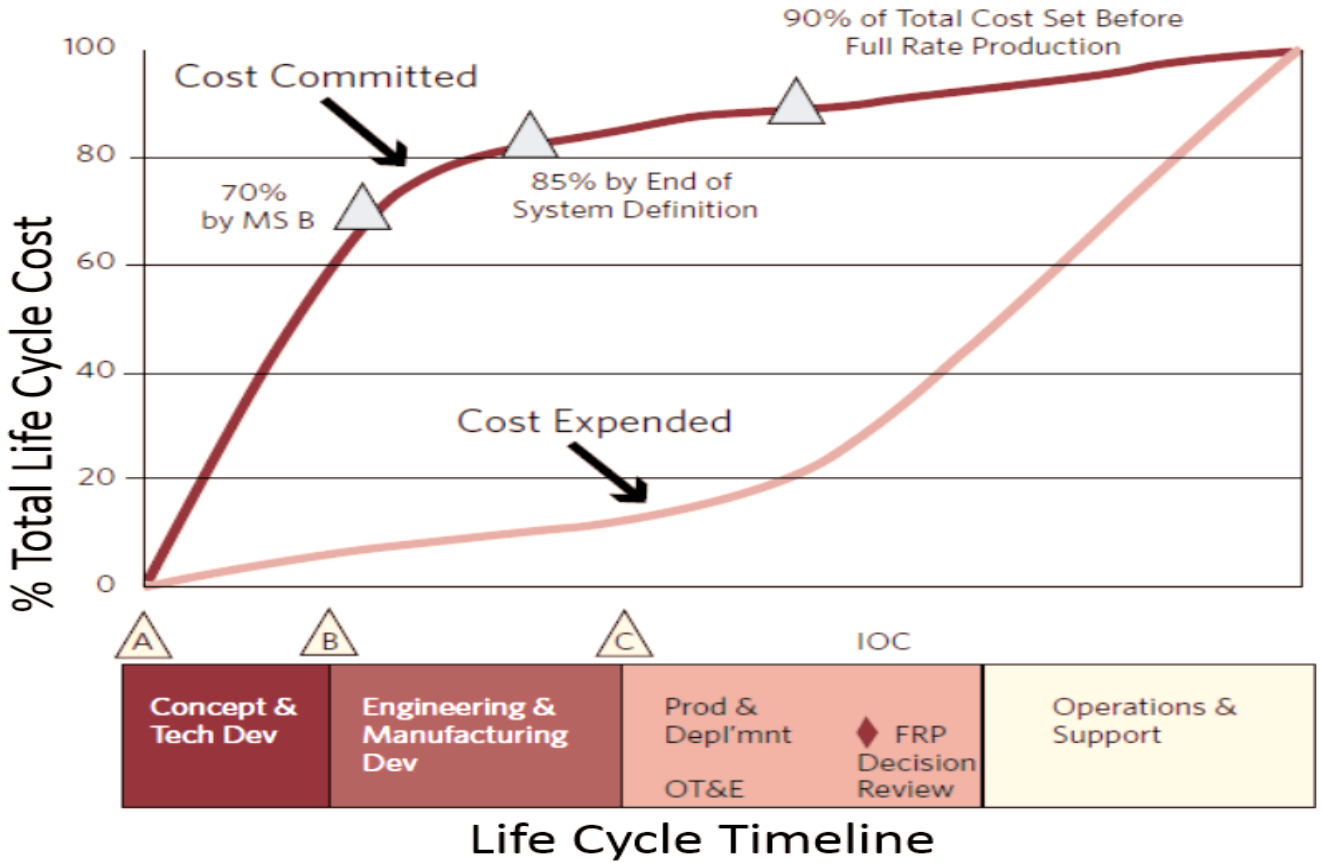
Instead of "development and sustainment", there is an acquisition community recognition that "initial development" and "continued development" are more accurate terms.

DSB #6: Software Sustainment
Current State 1

Software is a crucial and growing part of weapon systems/national security mission

Software essentially never dies - it requires DoD to update continuously and indefinitely until retirement

Operations and sustainment costs generally comprise 70-90% of the total lifecycle costs



DSB #6: Software Sustainment
Current State 2

Software sustainment is essentially a continuation of the original development.

- In general, the software development environment needs to be replicated for sustainment
- Sustainment organizations are not part of the system program offices from the inception of the project; thus, there contributions may be minimized.

Currently, there are several efforts to develop mechanisms to improve the transition to a software factory and CID.

RFPs do not have the basic elements of the software framework supporting the software factory

DSB #6: Software Sustainment Obstacles 1

Recommendation Area	DoD Obstacles	Industry Obstacles
The amount of the future software sustainment work will increase	The future on the sustainment of software systems will be more connected and more software complex	Modern technologies (e.g., Artificial Intelligence (AI) and Machine Learning (ML)) are increasingly important parts of a broad range of defense systems, including autonomous systems, and will further complicate the challenges for initial and continuous development of software for industry & DoD
The sustainment organizations have ecosystems that currently exist and must be considered	Approximately 30 DoD software sustainment organizations successfully respond to a range of customer needs and deliver critical software updates and enhancements, often under the intense schedule pressure of wartime operations, to deliver critical warfighter capability	RFPs for acquisition programs entering risk reduction and full development specify the basic elements of the software framework supporting the software factory.

DSB #6: Software Sustainment Obstacles 2

Recommendation Area	DoD Obstacles	Industry Obstacles
Contracting issues need to be resolved	Lack of agreement on data rights that satisfy both Government and Supplier needs.	Creative acquisition strategy approach to ownership of the Technical baseline

DSB #6: Software Sustainment Path Forward

Initiative	Action Plan
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 socialization of proposed contracting language • Conduct a set of pilot programs • Develop policies and guidance
Develop an understanding of the current and future sustainment organizational ecosystems to ensure the effective transfer of the software factories.	• NDIA workshop with government and contractor personnel • Generation and socialization of effective transfer mechanism • Conduct a set of pilot programs • Develop polices and guidance

DSB #7: IV&V for Machine Learning

Independent Verification and Validation (IV&V) for Machine Learning (ML)

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 Department must invest to build a better posture in this critical technology. Under the leadership and immediate direction of the USD(R&E), the Defense Advanced Research Projects Agency (DARPA), the SEI FFRDC, and the DoD laboratories

- should establish research and experimentation programs around the practical use of machine learning in defense systems with **efficient testing**, **independent verification and validation**, and **cybersecurity resiliency and hardening** as the primary focus points
- They should establish a machine learning and autonomy data repository and exchange along the lines of the U.S. Computer Emergency Readiness Team (US-CERT) to collect and share necessary data from and for the deployment of machine learning and autonomy
- They should create and promulgate a methodology and best practices for the **construction**, **validation**, and **deployment** of machine learning systems, including **architectures** and **test harnesses**

DSB #7: IV&V for Machine Learning
Picture of Success (end-state) 1

Consensus government/industry IV&V framework for systems with ML that:

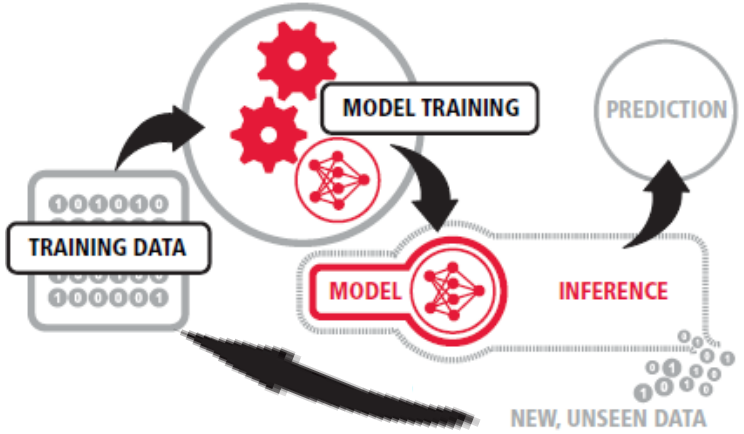
- Considers the full system context (not just the ML models) – a model-based inference engine is part of a larger software system designed to fulfill specific mission needs
- Adopts a risk-based methodology to support the Test and Evaluation (T&E) needs associated with ML in the system, using the mitigation of associated risks as a core part of the T&E process. (Motivated in part by DODI 8510.01)
- By identifying risks linked to ML model failures early in the system development process, we can take a range of actions to mitigate those risks throughout the system design, development, and sustainment lifecycle

T&E is a full lifecycle activity focused on mitigating risk of failing to meet operational needs

DSB #7: IV&V for Machine Learning
Picture of Success (end-state) 2

Open Data Sets - High data quality, quantity, availability, and traceability

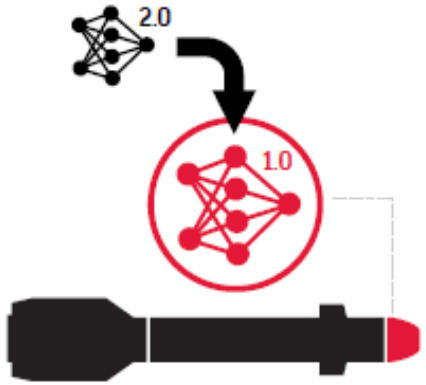
- Government data repository is accessible to industry and government. A governance model ensures that data sets are made available to all model developers - leveling the playing field , driving innovation, and increasing value to the Government
- New data is continuously collected by deployed systems, generating new training data sets that are published to the repository/exchange



DSB #7: IV&V for Machine Learning
Picture of Success (end-state) 3

Perpetual Upgrades - DevSecOps and associated CID supports continuous ML model updates

- ML models undergo continuous updates, so that their performance/accuracy stay aligned with a changing environment, evolving threats, availability of new data, etc.
- Continuous V&V methods tied to sensing of changes from models and environment
- A DevSecOps CONOPS for the construction, validation, and deployment of machine learning systems supports necessary ML model evolution *at the speed of relevance.*
  - We employ a CID approach for systems with embedded ML, enabled by a software Factory and a Continuous Delivery pipeline



Perpetual Upgrades

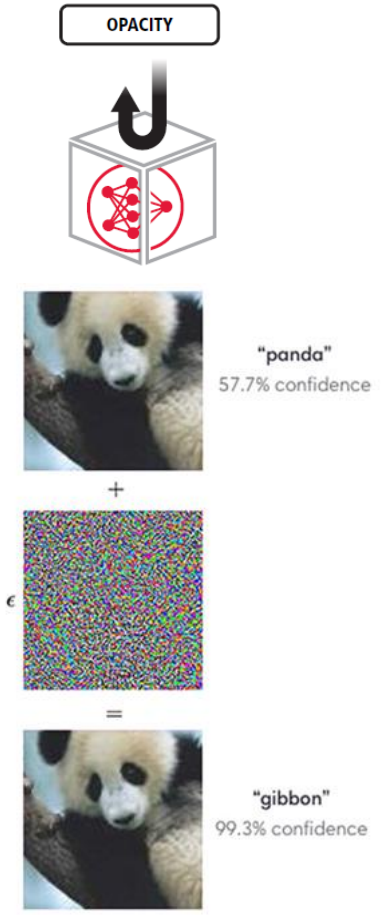
DSB #7: IV&V for Machine Learning Current State

Opacity - The "black box" nature of ML models combined with well-known sensitivities to the data sets used to develop (train) them raises many questions related to how we test them within their 'system context', and the level of trust we can place in them

Operational Risk - We train, test, and validate with data sampled from the "real/hidden" distribution, but operationally run models (inference) against examples from the entire population. If the training data does not adequately sample the real world, the model performs poorly (under-fitting, over-fitting). This represents operational risk

Data Availability - There is a general lack of availability of standard, shared, data sets for defense applications, and a general lack of data provenance and pedigree information

Security Risk - The potential for adversarial attack, for example, to disrupt the performance of an ML algorithm further exacerbates the operational risk concern



From: E. Ackerman. ***Slight Street Sign Modifications Can Completely Fool Machine Learning Algorithms.*** In IEEE Spectrum, August, 2017.

The 2016 Defense Science Board Summer Study on Autonomy discusses these challenges extensively and recommends that the DoD develop new strategies and approaches for T&E.

DSB #7: IV&V for Machine Learning Obstacles

Recommendation Area	DoD Obstacles	Industry Obstacles
Low TRL of test and evaluation techniques for "ML-in-the-system"		• Techniques for evaluation of Trust, Risk, Correctness, and other aspects of "ML-in-the-system" are still in early stages of R&D • Use of formal methods and other advanced techniques not yet established to the point where agreed-to procedures exist, that we can easily outline in a quality assurance process
Data access and availability	Concerns around data classification, data rights, and data ownership	• Lack of a common data repository and exchange • There is no standard approach to sharing data across industry

DSB #7: IV&V for Machine Learning Path Forward [1]

Initiative	Action Plan
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

DSB #7: IV&V for Machine Learning Path Forward 2

Initiative	Action Plan
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.

Summary

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

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

Acronyms

ACAT	Acquisition Category
ACWP	Actual Cost of Work Performed
AI	Artificial Intelligence
APIs	Application Programming Interfaces
ATDD	Acceptance Test Driven Development
ATO	Approval to Operate
BCWP	Budgeted Cost of Work Performed
BCWS	Budgeted Cost of Work Scheduled
CAPE	Cost Assessment and Program Evaluation
CDD	Capabilities Description Document
CDRL	Contract Data Requirements List
CEO	Chief Executive Officer
CI/CD	Continuous Integration / Continuous Delivery
CID	Continuous Iterative Development
CONOPS	Concept of Operations
COTR	Contracting Officer's Technical Representative
COTS	Commercial Off-the-Shelf Software
CP	Competitive Prototyping
CSB	Configuration Steering Boards
CV	Cost Variance
DARPA	Defense Advance Research Project Agency
DAU	Defense Acquisition University
DAWDF	Defense Acquisition Workforce Development Fund
DB	Database
DevSecOps	Development / Security / Operations
DIB	Defense Industrial Base
DoD	Department of Defense
DoDAF	DoD Architecture Framework
DODI	Department of Defense Instruction
DSB	Defense Science Board
EAC	Estimate at Completion
ETC	Estimate to Complete
EVM	Earned Value Management
FFRDC	Federally Funded Research and Development Center
FOC	Full Operational Capability
FOSS	Free and Open Source Software
FRP	Full Rate Production
GOTS	Government Off-the-Shelf Software
GPUs	Graphics Processing Unit
IATO	Interim Approval to Operate
ICE	Independent Cost Estimate
IDIQ	Indefinite Delivery / Indefinite Quantity
IMP	Integrated Master Plan
IMS	Integrated Master Schedule
INCOSE	International Council on Systems Engineering
IOC	Initial Operational Capability
IP	Intellectual Property
IPT	Integrated Product Team
IR&D	Internal Research and Development
ISO	International Organization for Standardization
IV&V	Independent Verification and Validation
ML	Machine Learning
MOSA	Modular Open Systems Approach
MS	Milestone
MTBF	Mean Time Between Failures
MTTA	Mean Time to Attack
MTTD	Mean Time to Detect
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair or Mean Time to Resolve
MVP	Minimum Viable Product
NDAA	National Defense Authorization Act
NDIA	National Defense Industrial Association
NRO	National Reconnaissance Office
NVP	Next Viable Product
OT&E	Operational Test and Evaluation
PARCA	Performance Assessment and Root Cause Analysis
PDR	Preliminary Design Review
PEOs	Program Executive Officers
PMB	Performance Measurement Baseline
PMO	Program Management Office
PM	Project Manager, Program Manager, Product Manager
PSM	Practical Software and Systems Measurement
p_{WIN}	win probability
RFP	Request for Proposal
ROI	Return on Investment
RTA	Run Time Assurance
SAFe	Scaled Agile Framework
SEI	Software Engineering Institute
SERC	Systems Engineering Research Center
SETR	Systems Engineering Technical Review
SIL	Software Integration Laboratory
SLOC	Source Lines of Code
SME	Subject Matter Expert
SOO	Statement of Objectives
SOW	Statement of Work
SRR	System Requirements Review
SV	Schedule Variance
T&E	Test and Evaluation
TPUs	Tensor Processing Unit
TRL	Technical Readiness Level
US-CERT	U.S. Computer Emergency Readiness Team
USD(A&S)	Under Secretary of Defense / Acquisition and Sustainment
USD(R&E)	Under Secretary of Defense / Research and Engineering
WBS	Work Breakdown Structure
WG	Working Group

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:

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

For More Information, contact the NDIA working group leads:

Joseph P. Elm
L-3 Technologies
Chair, NDIA Systems Engineering Division
Email: joseph.elm@L3T.com
Phone: 412-967-7295

Geoff Draper
Harris Corporation
Vice-Chair, NDIA Systems Engineering Division
Email: gdraper@harris.com
Phone: 321-727-6890