National Defense
Industrial Association

Practical Software &
Systems Measurement

International Council
on Systems Engineering

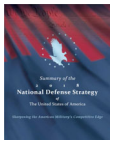
Continuous Iterative Development (CID) Measurement Framework v2.0

Cheryl Jones - US Army Futures Command - DEVCOM AC
Geoff Draper - L3Harris Technologies

NDIA SED - April 2021

The Evolution of SW Acquisition in DoD

National Defense Strategy 2018



"Performance at the speed of relevance" Streamline rapid, iterative approaches

Hon. Ellen M. Lord
USD (A&S)

Jeff Boleng
Special Ass't for SW Acquisition

2018 NDAA
(Sec. 872)
• DIB analyze SW regs

2019 NDAA
(Sec. 868)
• Implement DSB recommendations

Defense Innovation Board (DIB)
Software Acquisition and Practices (SWAP)
https://innovation.defense.gov/software/

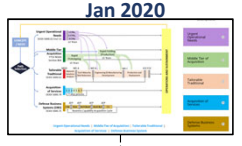
DIB SWAP
May 2019

- 3 Themes
- 4 Lines of Effort
- Top 10 recommendations
- DevSecOps
- Implementation plans

2020 NDAA
• Implement DIB recommendations

SW Acq Pathway
Interim Policy and Procedures
Jan 2020

Adaptive Acq Framework
Guidance, training
https://aaf.dau.edu/aaf
Jan 2020



CY17 | CY18 Q1 | CY18 Q2 | C18 Q3 | CY18 Q4 | CY19 Q1 | CY19 Q2 | CY19 Q3 | CY19 Q4 | CY20 Q1 | …

DSB SW
Feb 2018

7 recommendations
- SW factories
- CID
- Risk reduction, metrics
- Current & legacy programs
- Workforce
- IV&V for machine learning

Defense Science Board
Design and Acquisition of Software for Defense Systems
https://dsb.cto.mil/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf

Aug 2018

NDIA/INCOSE/PSM
CID WG
Mar 2019

http://www.ndia.org/divisions/systems-engineering/studies-and-publications

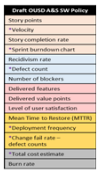
Industry recommendations for Implementing DSB findings

DoDI 5000.02 (1/23/20)
Operation of the Adaptive Acquisition Framework

https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500002p.pdf

SW CID Measures

Draft A&S SW Metrics



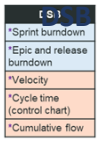
CID Measurement Framework
PSM/INCOSE/NDIA
Jan 2020

- Information Needs
- Measurable Concepts
- Indicator Specifications
- Guidance

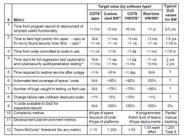
http://www.psmsc.com/CIDMeasurement.asp

Metrics

DSB



DIB



Industry Surveys (NDIA, INCOSE, PSM)

PSM User Group
CID Workshop
Sep 2019

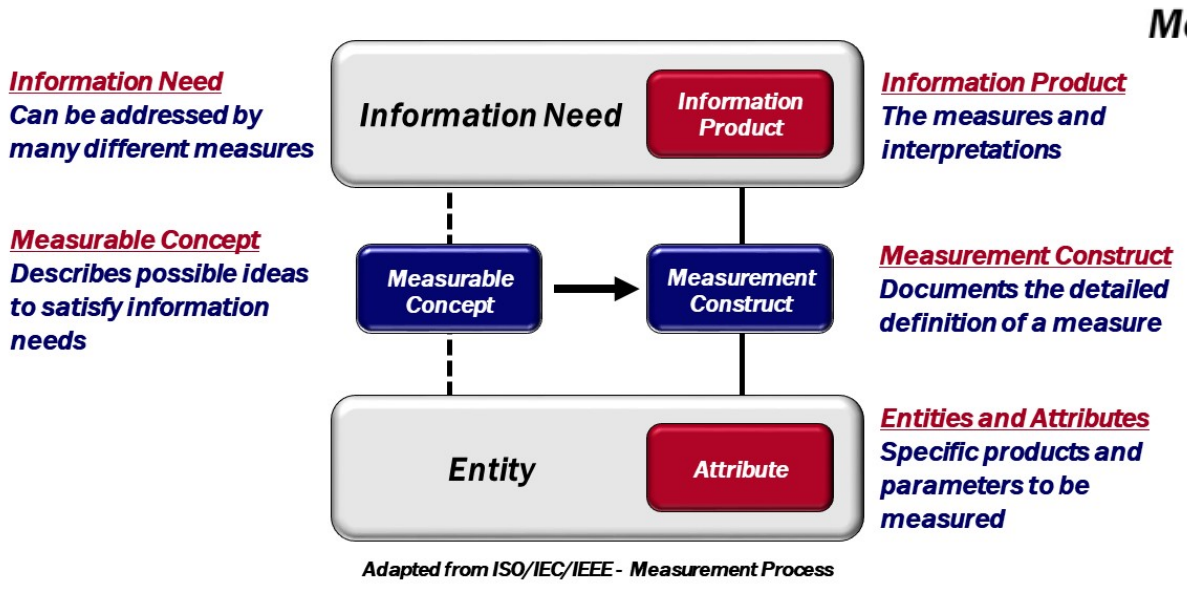
PSM Measurement Methodology

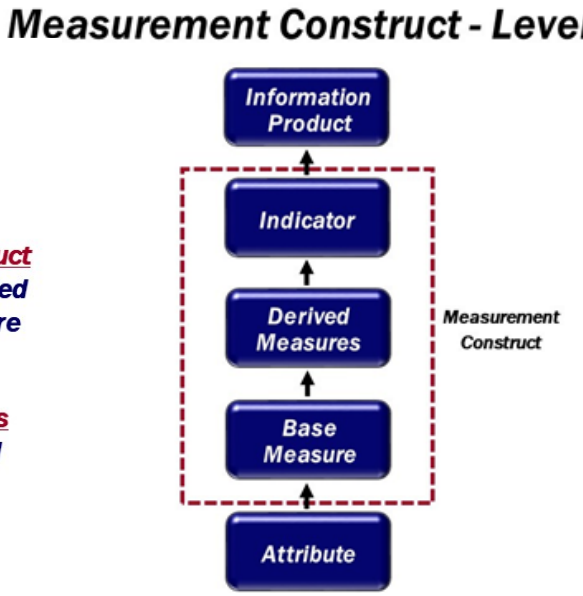
- **A collaborative decision environment founded on objective information and open communications enables the measurement results to positively impact program objectives**
- **Practical Software and Systems Measurement (PSM) principles are used to define relevant measures**
  - What you measure is driven by what you need to know - information needs
  - Measurement definitions and methods are determined by program and/or enterprise processes - process integration
  - The environment you work in drives how the measurement results are interpreted - decision context
  - Action must be taken to realize any benefit from measurement - fact based decision making

See Part 1, Section 5 for more information

Information Needs

- **Based on objectives and issues from the team, program, and enterprise levels**
  - Objective - a project goal or requirement
  - Issue - an area of concern that could impact the achievement of an objective, including risks, problems, and lack of information





See Part 1, Section 4 for more information

Information Needs, Categories, and Measures ICM Table (Excerpt)

Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
Schedule and Progress	Work Unit Progress (team, product) Milestone Completion (enterprise)	Are story points delivered as committed? Are we still on track to deliver all story points per roadmap? (on plan)	Are features/capabilities delivered as committed? Are we still on track to deliver all features/capabilities per roadmap? (on plan) What are the features/capabilities at risk of not being completed as scheduled? Are all capabilites/requirements allocated to releases?	Are capabilities delivered as committed? Are we still on track to deliver all capabilities per roadmap? (on plan) What are the capabilities at risk of not being completed as scheduled?	**Burndown Committed vs. Completed Velocity**
	Work Unit Progress		Did we deliver expected capabilities / features? Is the roadmap still valid?	Is the user satisfied with the delivered products? Do they provide the desired functionality when needed?	**Feature or Capability Implementation**
	Work Unit Progress		Is the integration and test progress proceeding as planned?		Test Progress

See Part 1, Section 7 for more information

CID Work Decomposition

Term	Synonyms	Description
Continuous Iterative Development (CID)	Agile, DevOps, DevSecOps, SAFe	A method of managing development, testing, and release of software, or systems, to continually, or iteratively, provide working functional systems of increasing capability to internal and external customers.
Roadmap		A high-level description, with text and visual, that maps out the vision and direction of product offerings over time. It describes the goals and capabilities of external releases. Dependencies between features/capabilities might be visualized. Relevant milestones, e.g., large-scale projects that interact with the product offerings, might be included.
Capability	Epic, Mission Requirement, Objective	Higher-level solution typically spanning multiple releases. For DoD, these may be reflected by a Capability Needs Statement (CNS) or JCIDS capabilities. Capabilities are made up of multiple Features to facilitate implementation.
Feature		A service or distinguishing characteristic of a software item (e.g., performance, portability, or functionality) that fulfills a stakeholder need and includes benefit and acceptance criteria within one release. Features are used to complete capabilities and are comprised of multiple Stories (or tasks, use cases, etc.). In some contexts, the term feature might also refer to software systems (capability-level scope) that ingest data, process data, and deliver a certain product/output to the stakeholders.
Story	Use cases	**User Story.** A small desired behavior of the system based on a user scenario that can be implemented and demonstrated in one iteration. A story is comprised of one or more tasks. In software development and product management, a user story is an informal, natural language description of one or more features of a software system. User stories are written from the perspective of an end user or user of a system. **Use Case.** In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a user and a system (or between software elements), to achieve a goal. Use cases can be used in addition to or in lieu of user stories.
Story Points		A subjective value assigned by the developing team to a story to provide a relative measure of effort and complexity. Story points are a unit-less value: they are a scalar indicator of relevant complexity. Story points are generally not comparable across teams.
Task		Steps to be completed to satisfy a Story.

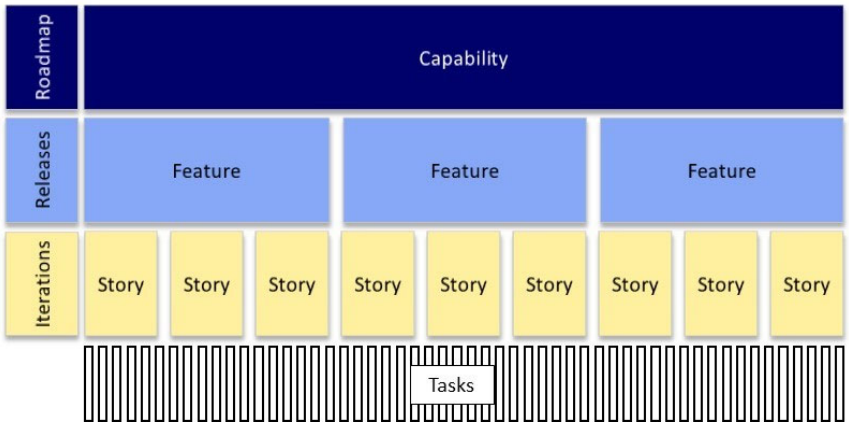


Figure 1: CID Work Decomposition

See Part 1, Section 2.1 and 3 for more information

Aligning the PSM framework and measures with DoD SW policy and enterprise improvement

- **Automated Test Coverage**
- **Burndown (Sprint / Release)**
- **Committed vs. Completed**
- **Cumulative Flow**
- **Cycle Time / Lead Time**
- **Defect Detection**
- **Defect Resolution**
- **Mean Time to Restore / Mean Time to Detect**
- **Release Frequency**
- **Team Velocity**

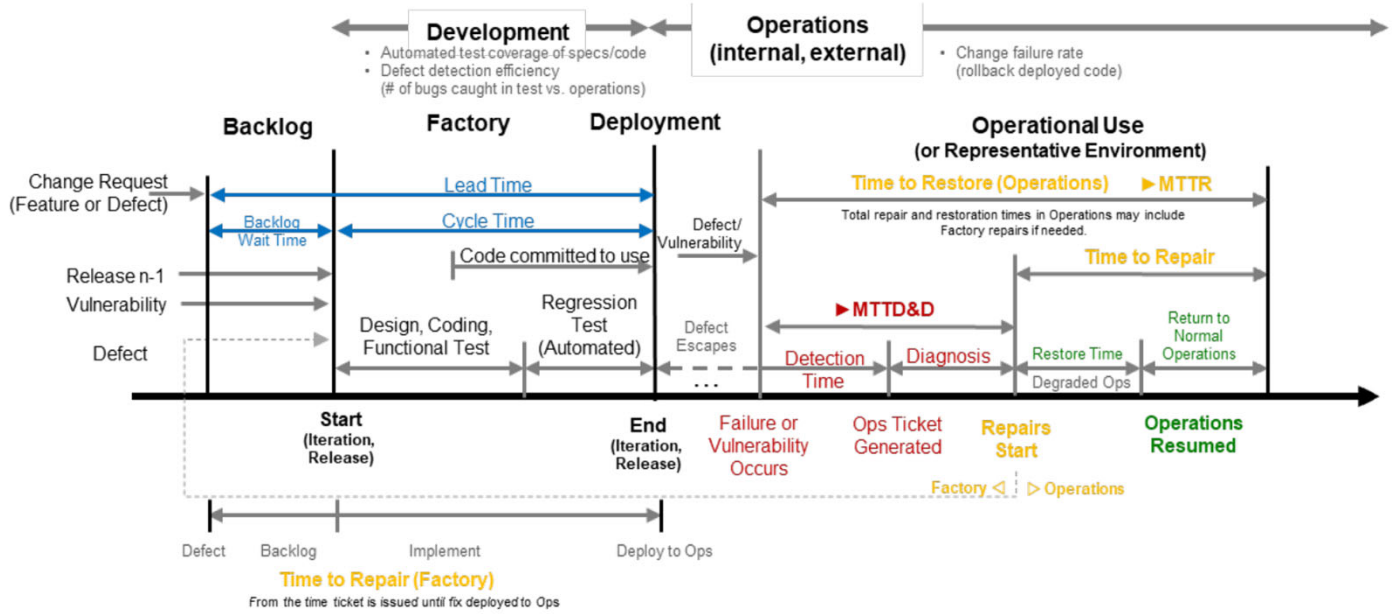


Figure 2: Measurement Context Diagram

See Part 1, Section 2.2 and 3 for more information

Release - Iteration - Defect Terminology

- **An ontology and definitions are provided**
- **Terms provide a foundation for consistency – different teams, programs, and enterprises may use different terms**
  - Synonyms provided

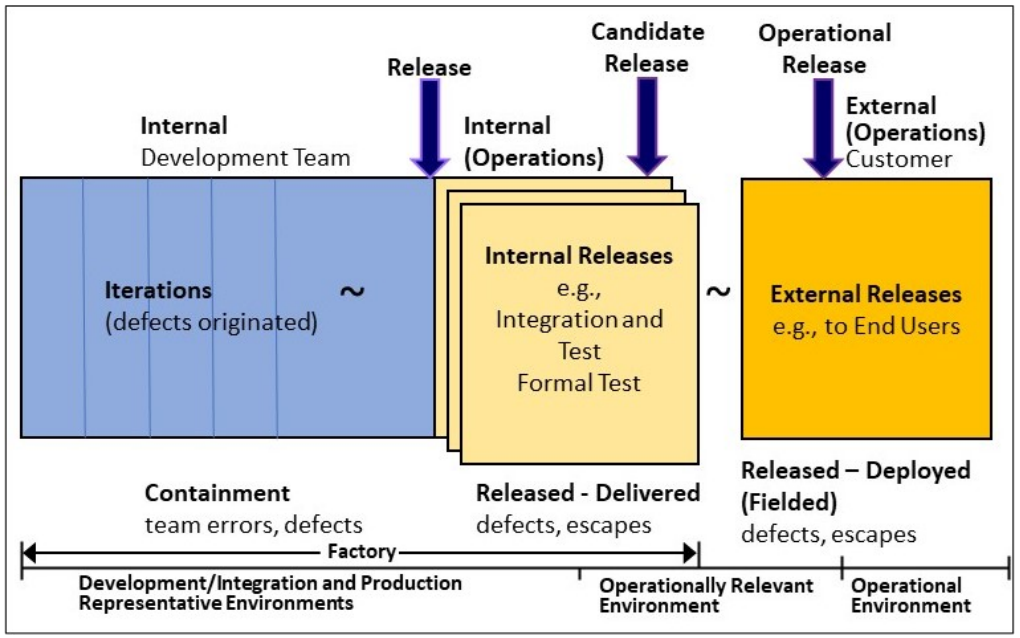
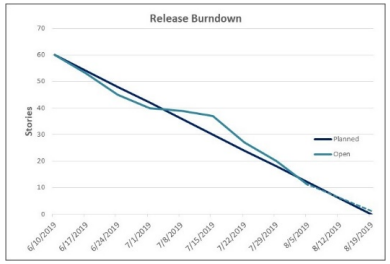


Figure 3

See Part 1, Section 2.3 and 3 for more information

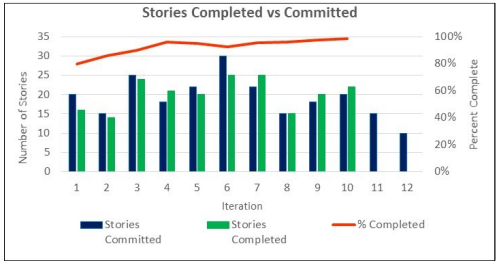
PSM CID Measurement Framework
Example Indicators - 1

Burndown



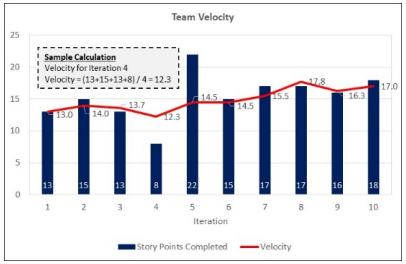
Is completion of work on plan?

Committed vs. Complete



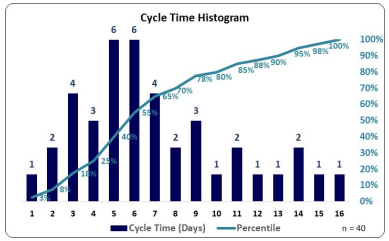
Is work delivered as committed?

Team Velocity



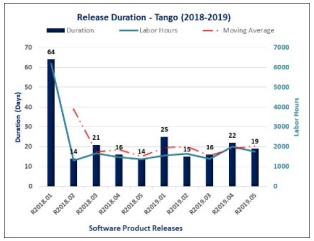
Is the team performing as expected?
How much work can be accomplished in future iterations?

Cycle Time / Lead Time



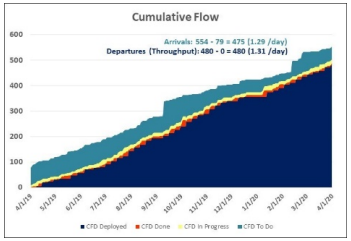
How long does it take to release a viable product?

Release Frequency



How long does it take to deploy a feature/capability?
What is the cadence or frequency for product release?

Cumulative Flow



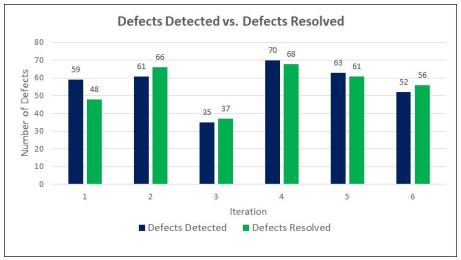
Is work flow moving through value stream?
Is the throughput of work predictable?

PSM Continuous Iterative Development (CID) Measurement Framework
http://www.psmsc.com/CIDMeasurement.asp

See Part 2, Section 8 for more information

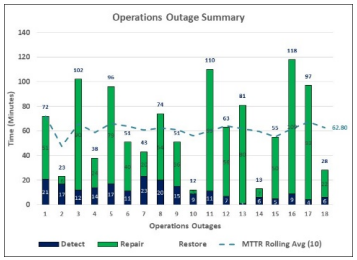
PSM CID Measurement Framework
Example Indicators - 2

Defect Resolution



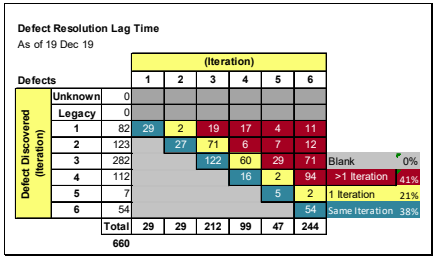
When are detected defects resolved? (aging)

MTTD / MTTR



How long does it take to detect and restore service incidents?

Defect Detection

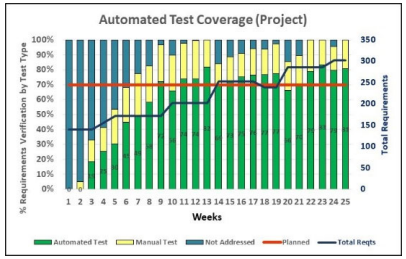


Defect Resolution Lag Time
As of 19 Dec 19

Defects			(Iteration) 1	2	3	4	5	6	
Defect Discovered (Iteration)	Unknown	0							
	Legacy	0							
	1	82	29	2	19	17	4	11	
	2	123		27	71	6	7	12	
	3	282			122	60	29	71	Blank 0%
	4	112				16	2	94	>1 Iteration 41%
	5	7					5	2	1 Iteration 21%
	6	54						54	Same Iteration 38%
	Total	660	29	29	212	99	47	244	

How many defects were released (saves, escapes)?

Automated Test Coverage



How much of the testing is automated?

PSM Continuous Iterative Development (CID) Measurement Framework
http://www.psmsc.com/CIDMeasurement.asp

See Part 2, Section 8 for more information

Sample PSM CID Measurement Specification

PSM Continuous Iterative Development Measurement Framework

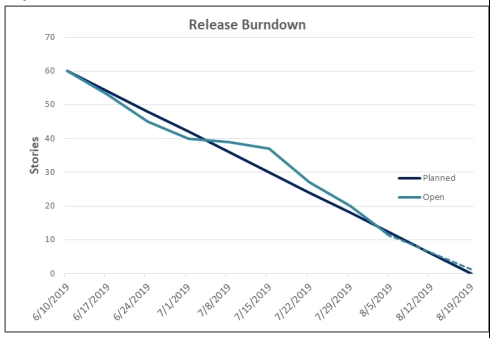
Developed and Published by Members of: PSM NDIA INCOSE

8.2 BURNDOWN (TEAM, PRODUCT, OR ENTERPRISE MEASURE)

Measure Introduction	
Description	Burndown is used to monitor completed work items (e.g., stories, features, capabilities) vs. planned work items for an iteration, release, or capability. Work items may include design, code, test and all supporting activities (e.g., requirements development, configuration management and quality engineering). Progress toward completing planned work is depicted graphically to provide an indicator of the likelihood of meeting planned goals.
Relevant Terminology	See Section 3: Ontology and Definitions.

Information Need and Measure Description	
Information Need	What is the status of the iteration, release, or capability? Will all the remaining committed work be completed as planned? What are the features/capabilities at risk of not being completed as scheduled? What are the trends in execution relative to plan?
Base Measure 1	Planned Work (integer scale) (e.g., Story Points/Features/Capabilities)
Base Measure 2	Completed Work (integer scale) (e.g., Story Points/Features/Capabilities)
Derived Measure 1	Open Work = Planned Work - Completed Work (e.g., Story Points/Features/Capabilities)

- Description
- Relevant Terminology
- Information Need
- Base Measures
- Derived Measures

Indicator Specification	
Indicator Description and Sample	In Figure 1, the teal line represents the number of open stories over time, while the blue line indicates the planned burndown. This chart shows a 2-month release, with weekly increments, where stories are completed.  **Figure 1: Release Burndown** At release planning, work items representing 60 stories were committed. While little progress was made during the first week to a planned training event, the teams recovered and is projected to complete the planned work by the end of the release.
Analysis Model	At the team level, the focus is generally on stories or story points open through the iteration. Is the team completing the committed work items? Are they significantly behind or ahead of the burndown plan? Are items blocked? What is the likelihood of meeting the commitment on time? Can additional backlog stories be brought into the iteration? Are teams improving execution over time? At the product level, the focus turns to features or capabilities across releases. At the enterprise level, the focus is generally on capabilities for external releases.
Decision Criteria	At the team level, lack of progress (e.g., not reducing open story points at all over several days) and variances from the plan (e.g., 5%) should be reviewed for action by the team. Data is generally not shared externally to the team. At the product level, variances of over 10% are reviewed for causes of roadblocks and consideration of replanning.

- Indicator Description and Sample
- Analysis Model
- Decision Criteria
- Additional Analysis Guidance
- Implementation Considerations

See Part 2, Section 8.2 for more information

Additional Information	
Additional Analysis Guidance	Use this metric with the velocity metric and other work unit progress metrics (e.g., test progress, cumulative flow). The velocity metric supports the planned story points for each iteration. The actual completed story points from the iteration is an input to the velocity metric. Review with other work unit progress metrics may support an assessment of overall risk and may impact prioritization of work for future iterations. Consider bounds of estimated burndown based on historical performance, e.g., best case, worst case, Monte Carlo analysis.
Implementation Considerations	Some teams may use hours instead of story points (or may map story points to hours).

Additional Specification Information	
Information Category	Schedule and Progress
Measurable Concept	Work Unit Progress
Relevant Entities	Product
Attributes	Story Points, Features, Capabilities
Data Collection Procedure	At the team level, story points committed for each iteration are determined at the iteration planning meeting. This value is determined from the velocity metric. Based on the average velocity and other factors (e.g., vacations), the team commits to a number of story points for the next iteration. Work items (e.g., stories, tasks) are selected to match this commitment. Work items are closed when completed and meet their evaluation criteria, and burndown progress is updated daily. At the product level, the features and capabilities committed for each release are determined during release planning. Commitments may be replanned as work is completed and priorities change.
Data Analysis Procedure	For the team, Burndown is analyzed daily for progress/risk and at the end of each iteration to determine if the story points were delivered as committed. The final story points completed value is an input to the velocity metric. For the project, Burndown is analyzed periodically (e.g., monthly, quarterly, by release). For the enterprise, Burndown of capabilities for major events is analyzed.

- Information Category
- Measurable Concept
- Relevant Entities
- Attributes
- Data Collection Procedure
- Data Analysis Procedures

V2.0 - Additional Focus Areas

V2.0 Measurement Focus

- **Product Value (Part 2, Section 8.11)**
  - PSM Product Value Measurement (PVM) provides a scalable, and flexible approach to measuring product value from three stakeholder perspectives: user, acquirer, and supplier
- Additional **enterprise indicators** for current measures **(Part 2, Section 9)**
- **Software Assurance Measurement (Part 3, Section 10)**
  - Working Group evaluating appropriate security measurement
- **Technical Debt (Part 3, Section 11)**
  - **Technical debt** consists of design or implementation constructs that are expedient in the short term but that set up a technical context that can make a future change more costly or impossible. May be related to architecture, design, structure, duplication, test coverage, comments and documentation, potential bugs, complexity, or coding practices.
  - Excludes **mission debt** such as functional deficiencies, required capabilities or features that should have already been implemented but are missing (distinguished from a backlog of features not yet prioritized for implementation), or missing functionality or performance issues in COTS for a COTS-intensive system.
- **Prioritization**
  - Enhancements, product value, technical debt, mission debt, and security all need to be identified and prioritized in iterations and releases

Product Value Measurement

- **PSM Product Value Measure (PVM) provides a scalable, and flexible approach to measuring product value from various of stakeholders perspectives**
  - Product value can have different meanings for different stakeholders
  - Product value is based on satisfaction of multiple attributes or characteristics which are important to the stakeholders
- **This measurement approach can work for software, hardware, systems, projects, and services if pertinent stakeholders can agree on related attributes that need to be evaluated**
- **The PSM PVM depends on identifying the objective and key stakeholders of the measurement: The objective and stakeholders determine the product attributes that will be evaluated and their importance**
- **The measure determines the percentage of available score assigned to each attribute by the subject matter experts performing the measure**
  - The results can then be normalized to the desired units and scale and combined in terms of attribute categories, subject matter experts performing the measure, or even stakeholder interests
- **A common set of product attributes and common stakeholders are included as a starting baseline in the PSM PVM**
  - Related sets of attributes are grouped into attribute categories
- **Attributes can be added to the measure or tailored as needed to meet the objectives and stakeholders of the measure**

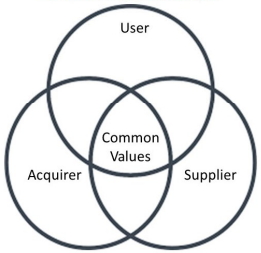
See Part 2, Section 8.11 for more information

Product Value Measurement

Product Value Depends on Perspective

- **Which stakeholders?**
  - User - Acquirer - Supplier
- **Based on which factors or dimensions?**
  - Categories - Attributes - Characteristics
  - Evaluation criteria
  - Timeliness (Impact of Delay)
  - Cost
- **For which measures?**
  - Qualitative - relative
  - Quantitative - tangible

Product Value Stakeholders



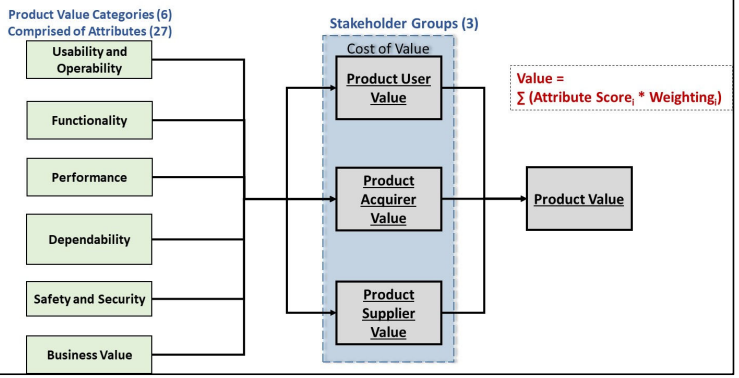
Establishes a scale by which stakeholders can communicate the importance of capabilities, or products of interest.

Example - Product Value Attributes by Category

Attributes for Performance Category

	Attribute Stakeholder User; Acquirer; Supplier =>	U	A	S
8	Does the system, product, or capability, perform to expected system measures of performance and effectiveness within expected, or contractual, system resource limitations?	U	A	S
9	Does the system behave gracefully when approaching resource limits such as large number of users or transactions or increased demand?	U	A	
10	Does the system, product, or capability provide the results within expected, or needed response time?	U		
11	Does the system meet or exceed the most important specified mission technical performance objectives, thresholds, or properties in an operational environment?	U	A	
12	Does the system provide sufficient margin for future growth in performance required to accommodate anticipated future mission needs?		A	S

Product Value Measurement Concept



Core Attributes of User Stakeholder Value

- **From the perspective of the end user or operator (ex. Pilot, infantry, maintenance technician, car driver, phone user, tablet user, etc.)**
- Usability and Operability
  1. Is the system, product, or capability, easy to use and operate?
  2. Are manpower, skills, and resources available to execute and maintain the system, product, or capability?
- Functionality
  3. Does the new/updated system ,or capability, work as intended, or required?
  4. Does the system, product, or capability, satisfy/improve mission needs?
  6. Does the system, product, or capability, align with the product roadmap?
  7. Are there operational or sustainment issues with the system, product, or capability?
- Performance
  8. Does the system, product, or capability, perform to expected system measures of performance and effectiveness within expected, or contractual, system resource limitations?
  9. Does the system behave gracefully when approaching resource limits such as large number of users or transactions or increased demand?
  10. Does the system, product, or capability provide the results within expected, or needed response time?
  11. Does the system meet or exceed the most important specified mission technical performance objectives, thresholds, or properties in an operational environment?
- Dependability
  13. Is the system, product, or capability, reliable and available when needed?
  14. Did you get the system, product, or capability, when you needed it?
  16. How easy does the system, product, or capability, recover operation from failure mode?
- Safety and Security
  19. Is the system, product, or capability, safe and secure to use?
  20. Does the system, product, or capability resist cyber and/or physical interruption, intrusion, spoofing, or degradation of its intended functionality and operation?
  21. Is the system, product, or capability, vulnerable to security attacks?

See Part 2, Section 8.11 for more information

PSM Product Value Measurement Terminology

Term	Definition
Product Attributes	Characteristics of the product, system, or capability that are important to a set of stakeholders.
Product Attribute Assessment	A value assigned by the SME to indicate satisfaction of the product to the attribute criteria. Each SME assigns a score for each attribute.
Product Attribute Weight	A value between 0 and 1 indicating importance of the attribute to stakeholders. The sum of all attribute weights for a product evaluated by a SME is 1.
Product Value	The satisfaction of Product Attributes that are important to the product Stakeholders. It is the aggregation of the Product Attribute Assessments.
Product Value Categories	Categories of Product Attributes related to product value. **Usability and Operability:** Ability of a product, system, or capability, to be easy to use and operate and effectively utilize personnel resources such as manpower and skills. **Performance:** The degree to which a system or component meets or exceeds technical requirements or delivery of capability that meet mission objectives with efficient system response and resource utilization measured or estimated under specified testing and / or operational environmental conditions. **Functionality:** Ability of a product, system, or capability, to provide or facilitate all the specified tasks and user objectives with the correct results and the needed degree of precision; and meet mission capability needs. **Dependability:** Ability of a product, system, or capability, to consistently performs it's intended functions over time, recover from any failure condition, be available and operable when needed. Includes availability, reliability, recoverability, maintainability, and maintenance support. **Safety and Security:** Ability of a product, system, or capability, to resist cyber and/or physical interruption, intrusion, spoofing, or degradation of its expected operation and functionality. **Business Value:** Ability of a product, system, or capability, to satisfy: customer initial and total cost targets; supplier contract performance, including delivery when promised; and supplier financial expectations throughout its lifecycle.
Stakeholder	A group, or individual, that has vested interested in the Product Value, i.e. User, Acquirer, Supplier. An organization may have multiple interests in a product and therefore they may belong to multiple stakeholder groups. For example: a company may be a supplier, acquirer, and user of a product. **User**: perspective of the end user or operator **Acquirer**: perspective of the purchasing organization, or buyer of the product **Supplier**: perspective of the company or organization that develops and provides the product to the Acquirer
Value Point	Unit of measure of product value based on evaluation of attribute criteria weighted by importance to the stakeholder.

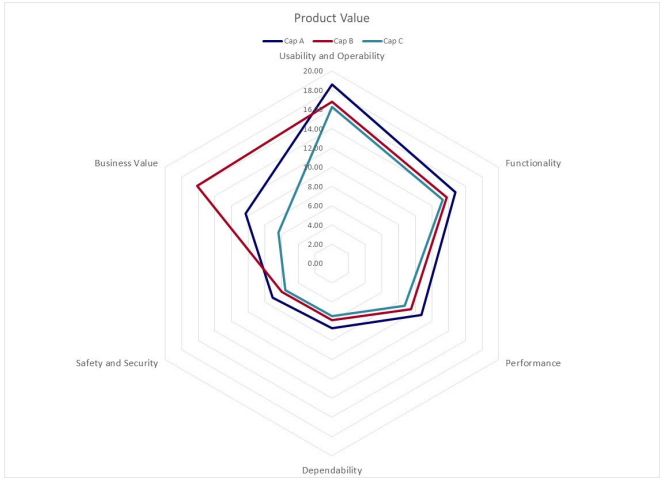
See Part 2, Section 8.11 for more information

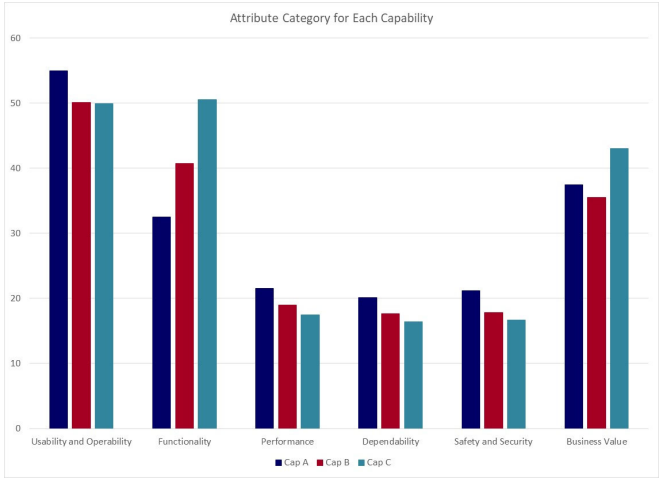
Example Use of PSM Product Value Measurement

- **Program Manager X will use the Product Value Measurement to determine which of 3 capabilities will be developed next**
  - This assessment will be repeated as the first capability approaches completion to prioritize the next capability to develop
  - The PM will look for the highest product value across the 3 capabilities but would like an emphasis on improving user experience
- **Objective: Determine the highest value capability out of a set of 3 candidate capabilities to be developed**
- **Ground Rules:**
  - There are 3 candidate capabilities. (A, B, and C)
  - The assessment will be performed by 3 SMEs (SME1, SME2, and SME3)
  - The SMEs are required to use attributes: 1-8, 14-16, 19-22. The weight for the others will be set to 0
  - The SMEs may set individual attribute weights as long as the total of attribute weighting will be 1.0. $\sum WTa = 1.0$
  - The scale used for the raw score will be max of 100 for all attributes.
  - To emphasize improving user experience attribute category weights will be used as follows: Usability = 0.25; Functionality = 0.20; Performance = 0.15; Dependability = 0.10; Security = 0.10; and Business Impacts = 0.20
  - A maximum target score of 100 will be used to normalize the final product value scores
  - The highest score will be selected for development
- **The product value total score will also be used to scale the progress measures to indicate how much value is incrementally being added for each increment**
- **Raw Scores of each capability generated by the 3 SMEs are in Additional Information section**

See Part 2, Section 8.11 for more information

Example: Results





	Category	WTc	Cap A Achieved	Cap B Achieved	Cap C Achieved
U	Usability and Operability	0.25	18.7	17.1	16.7
F	Functionality	0.20	14.7	13.2	13.0
P	Performance	0.15	9.9	9.3	8.7
D	Dependability	0.10	6.4	5.9	5.4
S	Safety and Security	0.10	7.1	6.0	5.6
B	Business Value	0.20	13.0	11.1	15.5
	Total	1.00	69.9	62.6	64.9

- 15 out of 22 attributes were used for the Product Value Measure
- Capability A has higher overall score and higher score in most attribute categories
- Capability C did have more Business Value than A
- Usability, Functionality, and Performance were the highest weighted categories and Capability A had higher scores in all three of these categories
- Capability A was selected for development even though Capability C scored higher for Business Value
- The Product Value of 69.9 was divided into the 4 planned increments to represent value provided by each

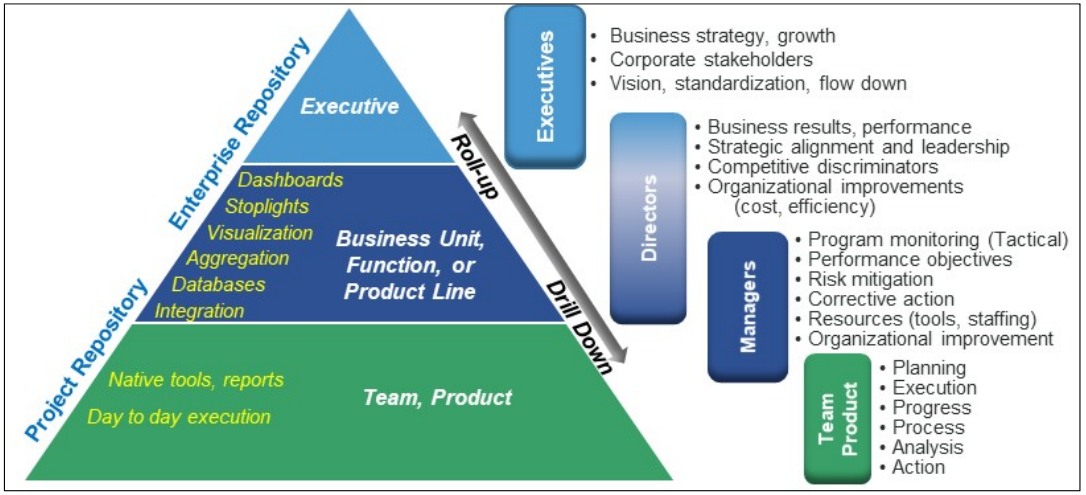
See Part 2, Section 8.11 for more information

CID Enterprise Measurement

- Team and product measures are used for insight and action at those respective levels, AND also to meet the higher-level needs of the enterprise, including:
  - Governance
  - Ensure effective performance on programs
- Measures may be summarized, aggregated, or transformed

Information Needs:

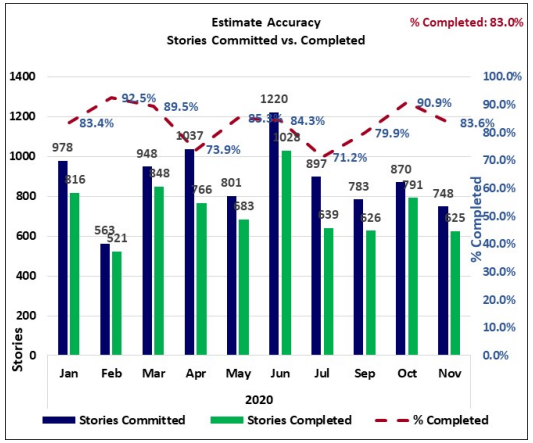
- How are our projects or products performing?
- Are we meeting commitments?
- What is the quality of products or services we deliver across the enterprise?
- Is productivity improving?
- How accurate are our estimates?
- What is the process efficiency for programs, businesses, or products?



See Part 2, Section 9 for more information

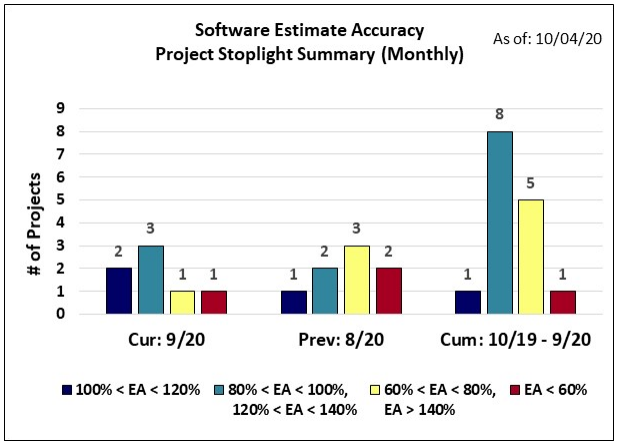
Enterprise Indicators - 1

Committed vs. Completed
Estimate Accuracy



How accurate are our estimates?

Are commitments being met consistently (plan vs. actual)?



Iteration Estimate Accuracy - by Stoplight Threshold Criteria

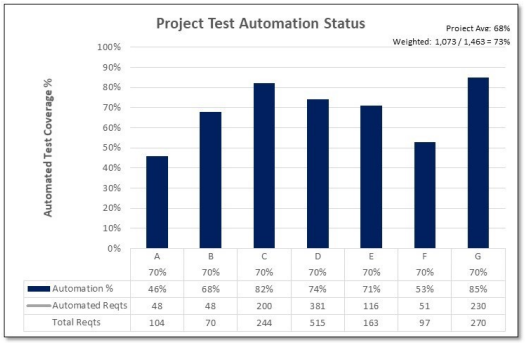
EA < 60%	60% < EA < 80%	80% < EA < 100%	100% < EA < 120%	120% < EA < 140%	EA > 140%

	Estimate Accuracy Stoplight Status (% Committed vs. Completed)												Total
Project	Oct-19	Nov-19	Dec-19	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20	Oct-19 Sep-20
Proj A	98.9%	130.4%	79.5%	94.8%	98.5%	84.7%	110.9%	93.7%	97.1%	98.1%	75.9%	106.3%	98.8%
Proj B							69.1%	100.0%	69.7%	105.6%	79.9%	81.1%	83.8%
Proj C	100.0%	100.0%	100.0%	100.0%	93.9%	90.2%	90.9%	109.3%	100.0%	98.1%	100.0%	100.0%	98.7%
Proj D	80.8%		66.2%	66.1%		78.3%	98.0%	96.4%	95.7%	82.6%		100.9%	85.0%
Proj E	79.0%	30.8%	152.8%	48.8%	119.2%	35.3%	55.8%	77.4%	98.9%	113.1%	57.9%	97.8%	77.5%
All	71.7%	93.7%	79.0%	84.6%	87.6%	80.7%	71.0%	75.6%	83.4%	72.4%	73.8%	74.0%	77.6%

See Part 2, Section 9.2 for more information

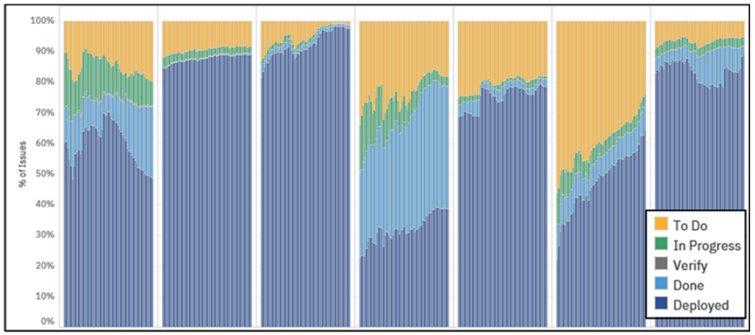
Enterprise Indicators - 2

Test Automation



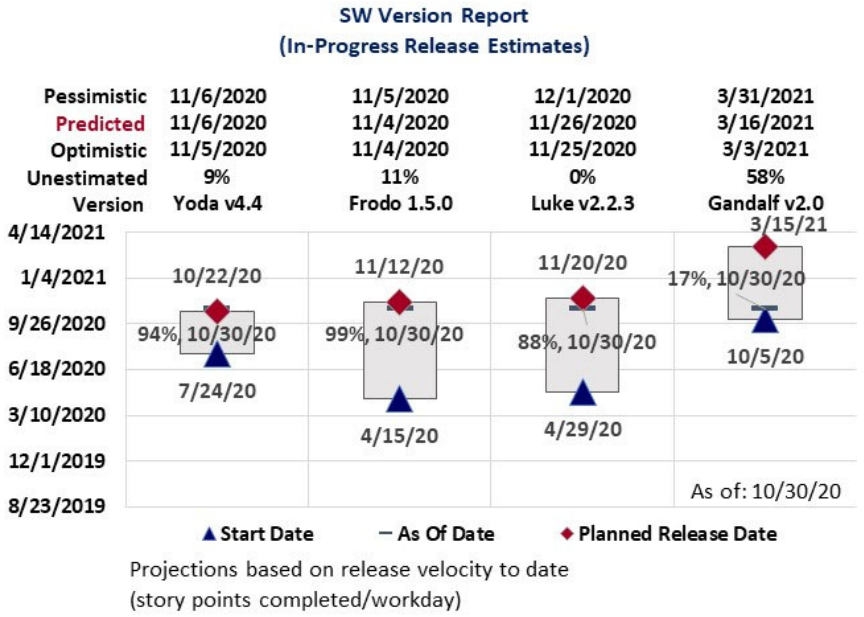
What is the extent of automated testing conducted across the enterprise's projects?

Cumulative Flow



Is current capacity keeping up with demand? Is the flow of work proceeding through the value chain?

Burndown



Projections based on release velocity to date (story points completed/workday)

How likely are we to deliver planned releases on time?

See Part 2, Section 9.2 for more information

Enterprise Indicators - 3

Defect Resolution

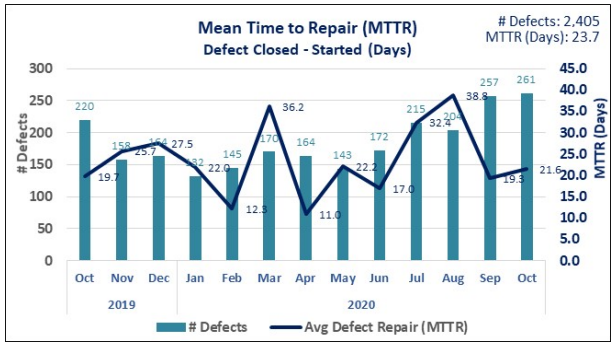
Defect Resolution Lag Time
As of 19 Dec 19

Defects			(Iteration) 1	2	3	4	5	6
Defect Discovered (Iteration)	Unknown	0						
	Legacy	0						
	1	82	29	2	19	17	4	11
	2	123		27	71	6	7	12
	3	282			122	60	29	71
	4	112				16	2	94
	5	7					5	2
	6	54						54
	Total	660	29	29	212	99	47	244

Blank	0%
>1 Iteration	41%
1 Iteration	21%
Same Iteration	38%

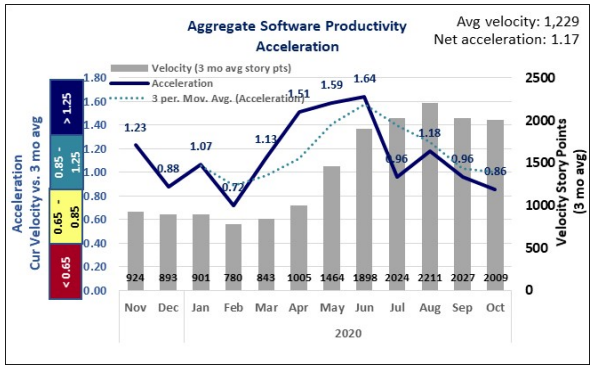
How long does it to take implement a feature or capability?

MTTR



How efficient are we at removing defects once found? How long does it take to restore service?

Velocity



Is productivity improving (more work completed per unit time)?

See Part 2, Section 9.2 for more information

Challenges for CID Measures at Enterprise Level

- **Data collection**
  - **Periodicity of Collection, Transformation, Visualization**
- **Roles, Authority, and Accountability**
- **Varying Reporting cadence**
  - **Between multiple programs and enterprise**
- **Inconsistent Project Measures**
  - **Aggregation and validation issues**
- **Enterprise Indicator Types**
  - **Summarize while capturing relevant details**

See Part 2, Section 9.1 for more information

Software Assurance Measurement

Emphasis on informing security considerations and measures throughout the life cycle

- **Vulnerabilities, weaknesses, defects**
- **Resources and execution of SW assurance verification and validation**
- **Software code screened and tested**

Table 1: Recommendations for Initial Software Assurance Measures

Measurement Concepts	Recommended Software Assurance Measures (Initial Priorities)
Identification and resolution of vulnerabilities and weaknesses	• Identification of vulnerabilities (CVEs) and weaknesses (CWEs) • Resolution of CVEs and CWEs • Patches delivered to burn down and close vulnerabilities
Security defect tracking	• Counts of security defects (open, closed) • Security defect attributes (e.g., severity, criticality) • Security defect containment (saves vs. escapes)
Quality and security testing coverage	• Percentage of code base screened for vulnerabilities and weaknesses (developed code and non-developmental items) • Security test coverage (code base, security controls) • Security test case status (passed, failed) • Trends in size of the attack surface

See Part 3, Section 10 for more information

Software Assurance Terminology

Term	Description
Software Assurance	Software Assurance (SwA) is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in the intended manner [CNSS 4009]. This ideal of no exploitable vulnerabilities is usually unachievable in practice, so programs must perform risk management to reduce the probability and impact of vulnerabilities and related weaknesses to acceptable levels.
Common Weakness Enumeration (CWE)	Common Weakness Enumeration (CWE™) is a community-developed list of software and hardware weakness types. It serves as a common language, a measuring stick for security tools, and as a baseline for weakness identification, mitigation, and prevention effort.
Common Vulnerabilities and Exposures (CVE)	Common Vulnerabilities and Exposures (CVE®) is a list of entries – each containing an identification number, a description, and at least one public reference – for publicly known cybersecurity vulnerabilities. CVE Entries are used in numerous cybersecurity products and services from around the world, including the U.S. National Vulnerability Database (NVD).
Common Attack Pattern Enumeration and Classification System (CAPEC)	Common Attack Pattern Enumeration and Classification (CAPEC™) is a community resource for identifying and understanding attacks. Understanding how the adversary operates is essential to cyber security. CAPEC helps by providing a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defenses.
Common Weakness Scoring System (CWSS)	The Common Weakness Scoring System (CWSS™) provides a mechanism for prioritizing software weaknesses in a consistent, flexible, open manner. It is a collaborative, community-based effort that is addressing the needs of its stakeholders across government, academia, and industry.
Common Vulnerability Scoring System (CVSS)	The Common Vulnerability Scoring System (CVSS) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. The numerical score can then be translated into a qualitative representation (such as low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes.

See Part 3, Section 10.1 for more information

Information Needs for Software Assurance (Excerpt)

Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures	Notes	Category and Priority
Product Quality Process Performance	Security - Safety Process Effectiveness			What is the (overall) compliance with the mission-critical Risk Management Framework (RMF) controls established for a program?	RMF Controls		SwA-M
Product Quality Process Performance	Security - Safety Process Effectiveness		How quickly can we detect a software assurance event or vulnerability? (Monitor, Detect) How quickly can we respond to a software assurance event? (Resolve, Deploy) How well you have designed the system to recover?	How rapid can the system recover to a known, secure state after an attack (Resiliency)? Is the system cyber-resilient? (Remove, recover)	Mean Time to Detect (MTTD) Mean Time to Restore/Recover (MTTR) Response Time Time to Patch Vulnerability Software assurance Vulnerability Lead or Cycle time	Also, a schedule and quality issue. Dependent on identified vulnerabilities (COTS issues or built-in). May be dependent on release process (e.g. Certifications and Assessments).	SwA-M
Product Quality	Security - Safety			Is my program protection planning adequate? Is there a software assurance strategy that maps to the Program Protection Plan?	Vulnerabilities Covered by Program Protection Vulnerabilities Removed Prior to Testing Code Passing Peer Review		SwA-M
Product Quality [Customer Satisfaction	Security - Safety		How much technical debt does the system have? What will it take to remove this technical debt?	How much technical debt does the enterprise have? What will it take to	**Technical Debt Actions (Written, Committed, Completed)** Effort/Cost to Resolve		TD-H

Publish Date: 20 January 2021 Version: v1.08 17

See Part 3, Section 12 for more information

Technical Debt

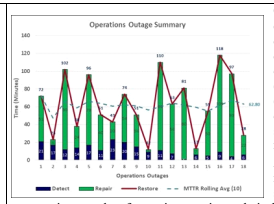
Technical debt consists of design or implementation constructs that are expedient in the short term, but that set up a technical context that can make a future change more costly or impossible.

- Debt from weaknesses in code constructs
- Debt from design and architecture decisions
- Debt from missing information items such as documentation shortfalls, missing information, IP issues.

11.2 Information Needs

Information needs related to technical debt include:

- How easy/difficult is it to update or refactor the design and code?
- Can the system architecture be expanded as the system continues to be developed and revised?
- When does it become too costly or take too long to maintain the design or architecture?
- How many defects are identified as technical debt?
- Is the documentation sufficient for user needs and for sustainability?
- When should identified technical debt be resolved, parts of the system replaced, or a new system started?
- What is the impact of this technical debt? Is it worth the investment and schedule to resolve it?

Mean Time to Restore (MTTR) / Mean Time to Detect (MTTD) (8.8)	 Operational services can be impacted if defects escape from development to operations. MTTR (restore) and MTTD (detect) are indicators of how quickly full service can be restored. Availability and reliability of deployed services, and impacts to operations, are key factors in assessing technical debt.

- **Example use of CID measurement framework**
- **Tools**
- **Methods**

See Part 3, Section 11 for more information

Recommendations for Adopting CID Measures

- **Define relevant team, program, and enterprise information needs**
  - Start by identifying the information needs that should be addressed - not the specific measures
- **Identify specific relevant measures to address identified information needs**
  - Define pilots for the measures, including product value
- **Develop leadership approach to review program and enterprise actions and decisions based on the measures**
- **Define how leadership will use relevant measures for fact-based decision making**
- **Consider guidance in CID Measurement Framework v2.0**

http://www.psmsc.com/CIDMeasurement.asp

Next Steps

- ☑ **Publish PSM v2.1 CID measurement framework - 15 April 2021**
  - Support adoption and use
- **Consider additional measures to address additional priority needs**
  - Size measures
  - Estimation and Cost Prediction
  - Draft measurement specifications for software assurance and technical debt
  - Additional measures for CID
  - Update Product Value Measure
  - Update all measures based on feedback from usage
- **Ongoing community participation to improve the PSM CID framework**
  - Join the PSM/INCOSE/NDIA WG (bi-weekly teleconferences)
  - Outreach and engagement with stakeholder groups (e.g., Security WGs)

See Part 1, Section 6 for more information

Acknowledgements

Acknowledgments

1.1 CONTRIBUTORS

Table 1: PSM CID Measurement Framework Editors

Editors	Organization
Cheryl Jones	Army Futures Command - DEVCOM AC
Geoff Draper	L3Harris Technologies / NDIA Systems Engineering Division
Bill Golaz	Lockheed Martin Corporation
Paul Janusz	Army Futures Command - DEVCOM AC

Table 2: Core Team Contributors and their Organization

Core Team	Organization
Rob Adams	Raytheon Technologies
Christopher Ashford	BAE Systems
Elizabeth Ashwood	Quantech Services
Jeff Boleng	OUSD (A&S) / Software Engineering Institute
Katherine Bradshaw	US Air Force
Sean Brady	OUSD (A&S)
Ian Brown	Galorath
Brad Clark	Software Metrics
Mark Cornwell	OUSD R&E
Steve Cox	Telecote Research
Holly Dunlap	Raytheon Technologies
Dennis Ebersole	Perspecta
William Hayes	Software Engineering Institute
Ronda Henning	L3Harris Technologies
Stephen Henry	Defense Acquisition University (retired)
Joe Jarzombek	Synopsys
Suzette Johnson	Northrop Grumman Corporation
Jonathan Kiser	The Boeing Company
William Luk	BAE Systems
Brian MacDonald	Raytheon Technologies
Jason McDonald	L3Harris Technologies
Lindsay Migala	US Air Force
Thomas Murphy	Silver Bullet
Andrea Nibert	Leidos
William J. Nichols	Software Engineering Institute
Ken Nidiffer	GMU
Cory Ocker	Raytheon Technologies
Tony Powell	York Metrics
Cari Pullen	US Air Force
Don Reinertsen	INCEPTUM
Garry Roedler	Lockheed Martin Corporation (retired) / INCOSE
Gene Rosenbluth	Northrop Grumman Corporation
David Rosenfeld	L3Harris Technologies
Larri Rosser	Raytheon Technologies
Benjamin Schumeg	Army Futures Command - DEVCOM AC
Ben Short	L3Harris Technologies
Robert Simmons	Raytheon Company
Matthew Stahr	US Air Force
Carol Woody	Software Engineering Institute / CERT
Robin Yeman	Lockheed Martin Corporation

Thank you to the many contributors from PSM, INCOSE, and NDIA that helped to develop the CID Measurement Framework!

For more information or to join our team contact:

Cheryl Jones
cheryl.l.jones128.civ@mail.mil

Geoff Draper
geoff.draper@l3harris.com

Acknowledgments

Table 1: Additional Contributors to the Report

Additional Contributors	Organization
Dr. Barry Boehm	USC / Systems Engineering Research Center (SERC)
Cherrie Brown	US Navy
Connie Bustillo	Lockheed Martin Corporation
John Catrone	Raytheon Technologies
Yvette Chamberlin	L3Harris Technologies
Kevin Chapman	L3Harris Technologies
Dr. Robert Charette	ITABHI Corporation
David Chesebrough	NDIA
Christopher Costello	Army Futures Command - CCDC Armament Center
Victoria Cuff	OUSD (A&S)
Kyle Davis	Quantech Services
James Doswell	DOD
Rick Dove	Paradigm Shift / INCOSE
Kim Elliott	Raytheon Technologies
Joseph Elm	NDIA Systems Engineering Division
Esma Elmazaj	L3Harris Technologies
Trevor Enos	US Air Force
Jen Garcia	Raytheon Technologies
Firas Glaiel	Raytheon Technologies
Nat Heiner	Northrop Grumman Corporation
Diane Juhas	Raytheon Technologies
Matt Kennedy	US Department of Treasury
Jo Ann Lane	SDSU
Jessica Li	Lockheed Martin Corporation
Phyllis Marbach	INCOSE
Shannon Moore	US Air Force
Greg Niemann	Lockheed Martin
Victoria Perez	US Air Force
Bernard Reger	Army Futures Command - CCDC Armament Center
Carmela Rice	OUSD (A&S)
Laura Rodas	Lockheed Martin Corporation
Gene Rosenbluth	Northrop Grumman Corporation
Forrest Shull	Software Engineering Institute
Ranjit Singh	Lockheed Martin Corporation
Roz Singh	Raytheon Technologies
Dan Strickland	Missile Defense Agency
Steven Verga	L3Harris Technologies
Marilyn Vickers	US Air Force
Gan Wang	BAE Systems