Practical Software and Systems Measurement Continuous Iterative Development Measurement Framework

Part 1: Concepts, Definitions, Principles, and Measures

Version 2.1
April 15, 2021

Developed and Published by Members of:

Practical Software & Systems Measurement

Product No.
PSM-2021-03-001

National Defense Industrial Association

International Council on Systems Engineering

Product No.
INCOSE-TP-2020-001-06

Editors:

Cheryl L. Jones
US Army
cheryl.l.jones128.civ@mail.mil

Geoff Draper
L3Harris Technologies
geoff.draper@l3harris.com

Bill Golaz
Lockheed Martin
willliam.h.golaz@lmco.com

Paul Janusz
US Army
paul.e.janusz.civ@mail.mil

PSM Product Number: PSM-2020-06-001
INCOSE Product Number: INCOSE-TP-2020-001-06

Copyright Notice:

CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

This report provides recommendations for the measurement of continuous iterative developments (CID). The report includes a Practical Software and Systems Measurement (PSM) CID measurement framework detailing common information needs and measures that are effective for evaluating CID approaches. The information needs address the team, product, and enterprise perspectives to provide insight and drive decision-making. The framework also identifies and specifies an initial set of measures that have been identified as being practical measures to address these information needs.

This guidance is intended to be used by team, program, and enterprise personnel who are implementing CID approaches, as a reference for common, practical measures that can be utilized. The measures a program or enterprise chooses to implement and collect will be tailored based on alignment with its information needs and objectives, so they may differ from those described here. The measures presented are intended to be tailored and adapted to the development approach and environment.

Version 1.05 detailed potential information needs and measures that are common to CID approaches, and an initial set of ten measurement specifications that were prioritized by user surveys as highest value. This Version 2.1 includes added material that has been researched and developed by the CID working group. The new materials include information on measuring:

- Product value (Part 2, section 8.11)
- Enterprise measurement (Part 2, section 9)
- Software assurance (Part 3, section 10)
- Technical debt (Part 3, section 11)

Part 1 of this report includes a series of diagrams and an ontology to describe the development approaches and terminology used. It also includes an "Information Category-Measurable Concept-Measures" (ICM) Table detailing potential information needs and measures for CID developments. Additional potential measures will be added in future releases, as described in Section 6, Next Steps.

For the highest priority measures, sample measurement specifications have been developed that detail the identified measures. These are included in a separate Part 2 of the paper, along with a discussion of how to use these measures for enterprise decision making. Part 3 of the paper separately extends the main CID paper with detailed information and guidance on Software Assurance and Technical Debt.

We invite your comments on this material, and your participation in future updates addressing additional measures and guidance.

This report is intended to be methodology and approach-agnostic and is written so that it may be adapted to organizational needs. Different methodologies and tools may use different terminology than defined in this report.

1. Front Material

The following sections provide overview information.

1.1 Background

A collaborative working group was established between Practical Software and Systems Measurement (PSM), the National Defense Industrial Association (NDIA) Systems Engineering Division, and the International Council on Systems Engineering (INCOSE) to develop a PSM measurement framework for Continuous Iterative Development (CID) in response to recommendations of the Defense Science Board (DSB) and Defense Innovation Board (DIB) studies.

Additionally, the U.S. Department of Defense (DoD) is making a transformational change in acquisition policy by redesigning the Defense Acquisition System, including the addition of a new Software Acquisition Pathway (Software Acquisition Pathway Interim Policy and Procedures, 2020). The general guidelines for this new acquisition policy are established in Section 800 of the 2020 National Defense Authorization Act. The pathway promotes Agile and DevSecOps and allow for upgradeable and timely fielding of software in a way that aligns with this CID approach. The measurement recommendations in this report provide a methodology to measure the Execution Phase of the Software Acquisition Pathway. These CID measures also apply to other non-DoD domains.

The most critical information needs and measures have been prioritized, based on a series of surveys with members of relevant NDIA, PSM, and INCOSE working groups. Additional measures will be specified, and revisions to the information needs will be included, as additional input is provided. This framework will be improved over time. We welcome your recommendations and comments.

1.2 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

Additional thanks go to the many additional colleagues who contributed to the development of the guide thorough participation in meetings, workshops and reviews.

Table 3: 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

2. Major Concepts

This PSM CID measurement framework provides guidance on information needs and measures from three perspectives: team, product, and enterprise. In many cases, the same base measures may be used, although aggregated to higher levels for product or enterprise needs. In other cases, different base measures may be used. The measurement specifications provide initial guidance on tailoring measures and indicators for these different perspectives and aggregation levels.

For CID, stakeholders include actual users of the system and software, as well as the development team, customer, and enterprise managers. The measures need to provide value to all stakeholders and inform diverse information needs.

One of the major issues with measures is ensuring that they provide information needed to support decision making and that they are used. A small set of measures should be tailored for each program and organization, focused on those needed for fact-based decision making. The measures should be regularly reviewed to ensure they are being used. If not, other measures may be required, or additional training may be required for decision makers on how the measures can be utilized.

A successful measurement program depends on establishing a clear context and operational definitions for the measures to be collected. Definitions can sometimes vary depending on the references and how measures are applied. The diagrams and definitions that follow provide the terminology used in this PSM CID measurement framework, in order to establish a common understanding, so that measures can be implemented and used consistently with community consensus.

2.1 CID Work Decomposition

Figure 1 contains a sample work decomposition approach for CID. This terminology will be used throughout this report and the associated ICM Table and measurement specifications.

Mission Requirements or Capabilities are the top level of user requirements. They are often documented in a roadmap. The roadmap is a top-level view of capabilities, which evolves over time as the CID process is performed. For DOD systems, the mission requirements may begin in the Joint Capabilities Integration and Development System (JCIDS), Capability Needs Statement (CNS), or an equivalent document. Capabilities are then decomposed into features which are then decomposed into stories, which may be decomposed into tasks.

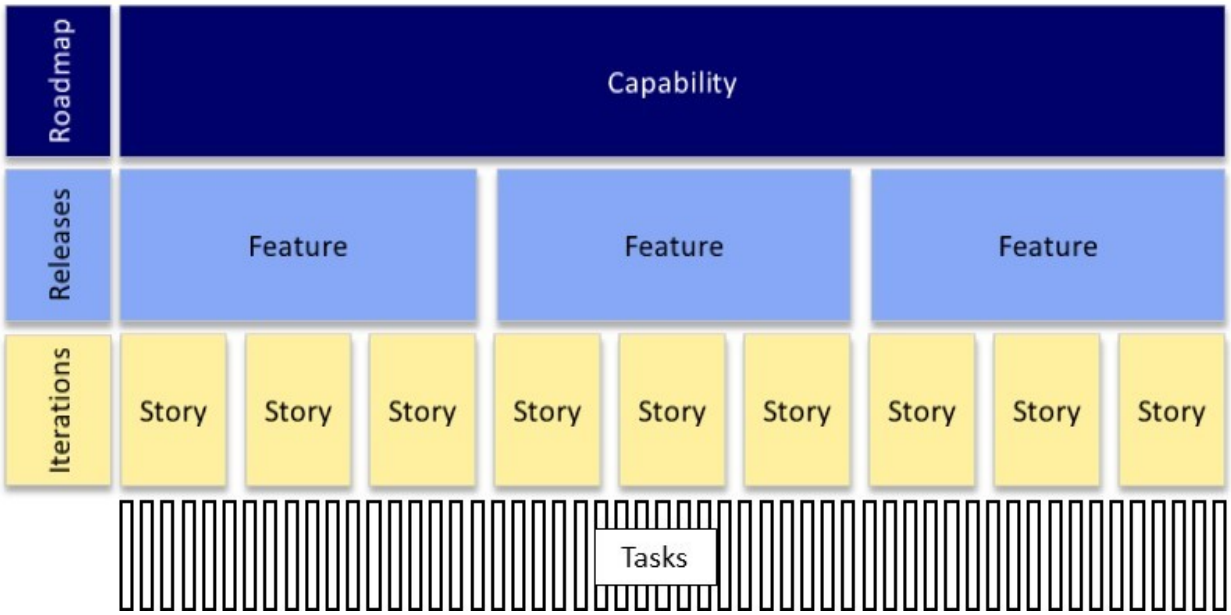


Figure 1: CID Work Decomposition

2.2 MEASUREMENT CONTEXT DIAGRAM

Figure 2 illustrates the context for common measures of continuous iterative development as they are defined and applied in the PSM CID measurement framework and measurement specifications. The diagram should be interpreted as a model supporting multiple iterations throughout development and operations. Although intended to be broadly applicable across a range of domains, adopters of the framework should further interpret, tailor, and apply these measures as applicable to their own business context.

Measurement may occur in each of many potential stakeholder environments, or across environments. Not all organizations will have all of these environments, as distinct entities. Different levels of sophistication of these environments may be used by different teams, for different levels of evaluation. Possible environments include:

- Development/Integration Environment(s)
- Production Representative Environment
- Operationally Relevant Environment
- Operational Environment

The enterprise generally focuses on actual measures from the operational environment. The team or product measures may begin in earlier environments, and focus on ensuring objectives will be met as the system is developed and sustained. Similar activities may be performed in different environments, with separate measures of effectiveness.

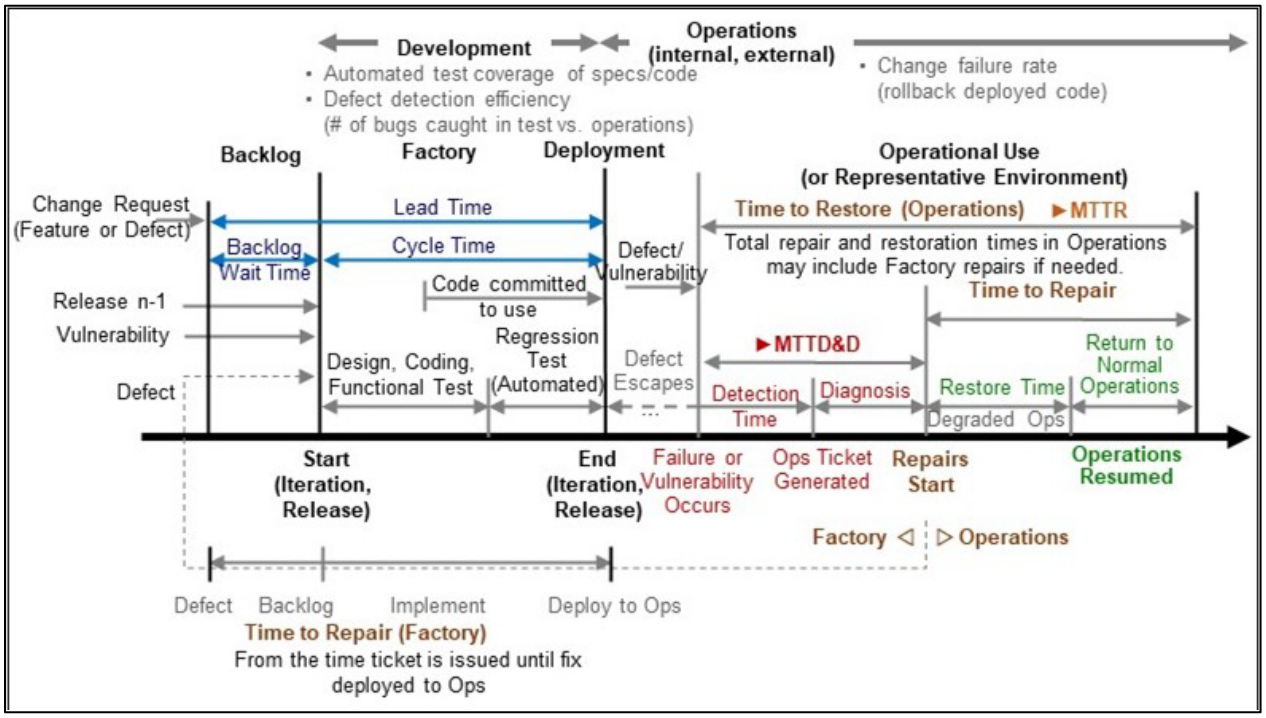


Figure 2: Measurement Context Diagram

Adapted from: https://limblecmms.com/blog/mttr-mtbf-mttf-guide-to-failure-metrics/

Major elements of this diagram for interpreting the context for candidate measures in the PSM CID measurement framework, emphasized by the bolded text labels, are described below. Additional details on individual measures are provided in the measurement specifications.

- **Backlog**: A collection of proposed work items to be implemented (see Section 3 for full description). Work items may include user needs (new or unfilled items) or defects from prior releases. Work proceeds for only those requests that are prioritized and accepted for implementation (committed work).
- **Factory**: Development proceeds through the Factory processes (requirements, design, implementation, test) for committed work and culminates with deployment. Work is planned and implemented iteratively (a recurring series of iterations and releases).
- **Operations**: Completed work from the Factory is Deployed in a new release to internal or external Operations, which may include a developer integration/test environment, end use Operations, or other intermediate operationally representative environments (e.g., operational test bed). The measures shown may be relevant to any or all of these environments. See Figure 3 for additional details on internal and external operations.
- **Rework**: The release(s) deployed may need to be updated to account for defects, security vulnerabilities, or other anomalies that affect the delivery of deployed services. Defects (e.g., trouble tickets) are issued for these requested changes. Operations may be able to continue in a degraded mode (e.g., workarounds, redundant paths) until full service is restored. Restoration time (Mean Time to Restore (MTTR)) includes the time to detect and diagnose the error (MTTD), and to implement and deploy repairs. The colors (Red, Brown, Green) in this figure indicate the transition from observation of the issue, to initiation of repairs, and to restoration of normal operations.

2.3 DEFECT TERMINOLOGY

Defect terminology may also change from one methodology or company to another. Defect terminology used in this PSM CID measurement framework is defined in the ontology in Section 3, consistent with Figure 3. Operationally representative environments can be either internal or external.

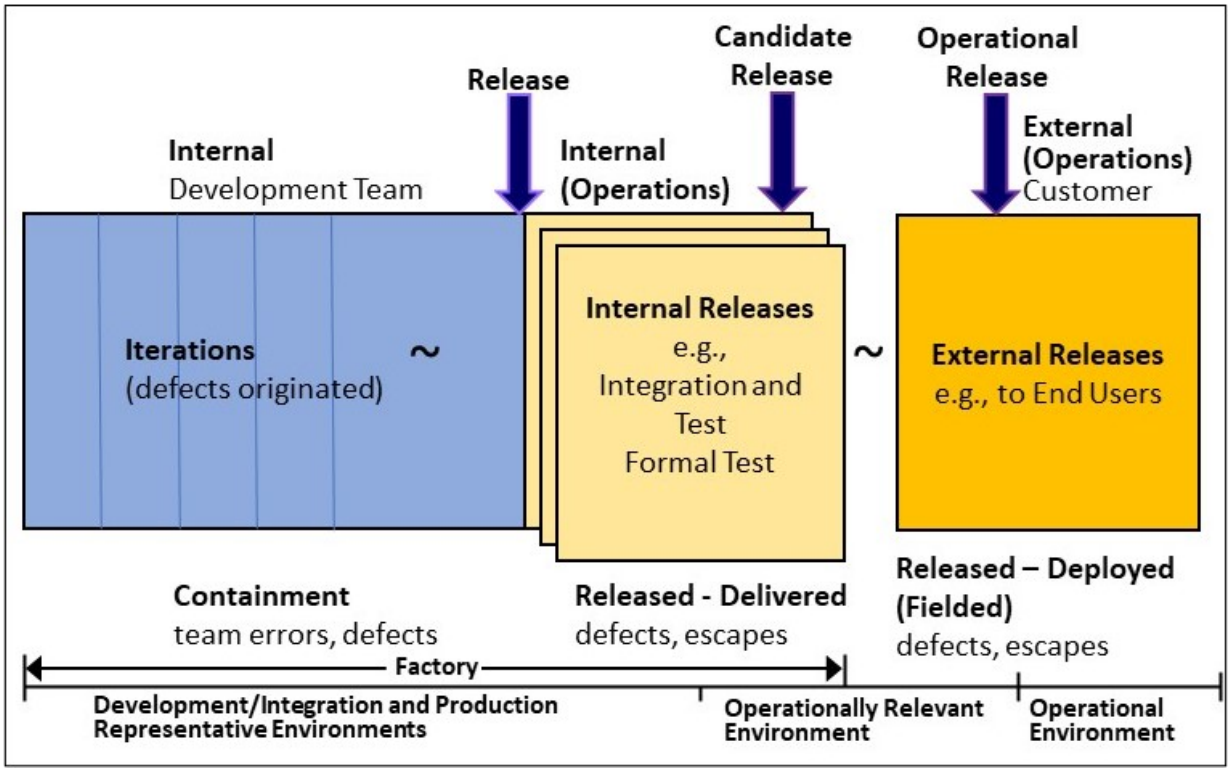


Figure 3: Defect Terminology

2.4 CID PROCESS

Figure 4 provides a conceptual depiction of the base measures that are collected for iterative releases and deployments to operations. There may be many iterations that are produced for internal use and continued development (for example v0.n, v1.n, v2.n in Figure 4). A subset of these are candidate releases that are available for external use (for example Release 1.0 in the figure), with a subset of these actually released for operational use (for example Release 2.0 in Figure 4). Some of these releases are assigned conceptual terms (MVP, NVP, MVCR) indicating the maturity of the product capability for early operational use; refer to Section 3 for descriptions.

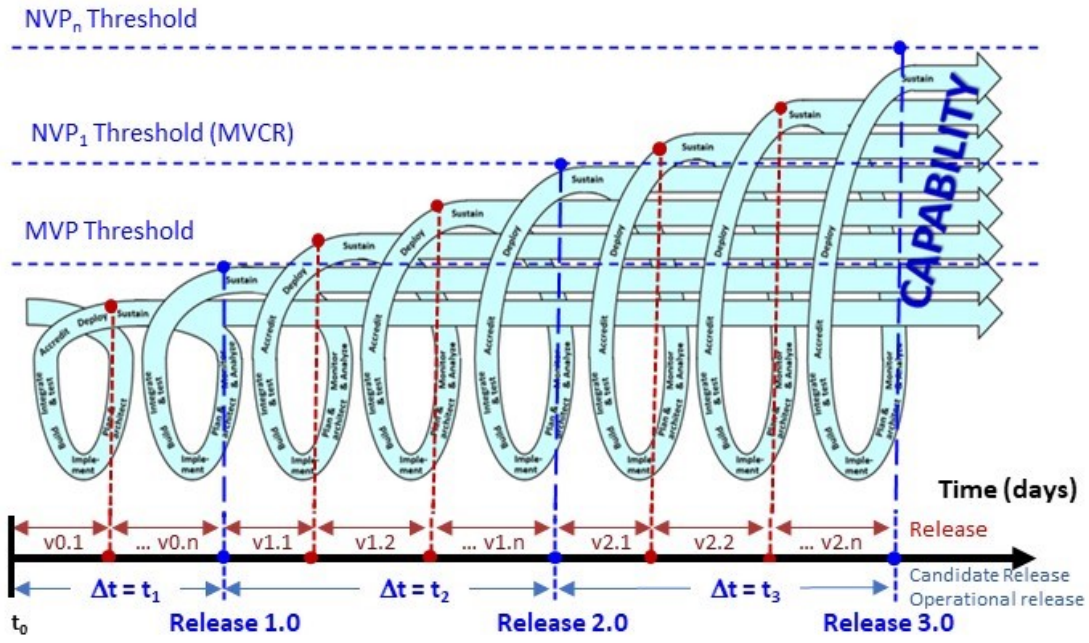


Figure 4: Continuous Iterative Development Process

3. Ontology and Definitions

The terms in Table 4 are used in the PSM CID measurement framework and specifications. Related terms are illustrated in figures 1, 2, and 3, and are grouped together in this section. The terms and definitions used here are drawn from several sources, including common industry best practices (defense and commercial), inputs from subject matter experts, DoD Software Acquisition Pathway policy and guidance, and DSB/DIB software acquisition reports. (See Bibliography for references.)

Table 4: PSM CID Measurement Framework and Specifications Terms

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.

Term	Synonyms	Description
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.
Cycle Time		The elapsed time from when work is put into progress until the time work has been completed.
Lead Time		The elapsed time from when work is identified, and a request is provided to the time the request has been satisfied. Note: The time the request has been satisfied is usually the same time the associated work is completed.
Backlog	Program Backlog Release Backlog	**Product backlogs** identify detailed user needs in prioritized lists. The backlogs allow for dynamic reallocation of scope and priority of current and planned software releases. The backlog contains new capabilities/features, changes to existing capabilities/features, defect fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome. Issues, errors, and defects identified during development and operations should also be captured in the product backlog to address in future iterations and releases. The development team works with the user community to decompose and prioritize the roadmap capabilities into product backlog entries. An **iteration backlog** is a list of the new stories, changes to existing stories, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome, within a near term iteration cadence. The iteration backlog contains a decomposition of product backlog entries into lower level items, for those prioritized for near-term implementation.
Problem Report	Defect Report, Discrepancy Report, Trouble Ticket	Identified issue with the product. Once approved for implementation, a Change Request, or Story, may be created, or the Problem Report may be used to track implementation. Service incidents in Operations are typically recorded in trouble tickets or equivalent.

Term	Synonyms	Description
Defect	Errors, Issues	A defect is a condition in a product (e.g. software, system, hardware, documentation) that does not meet its requirements or end-user expectation, causes it to malfunction or to produce incorrect/unexpected results, causes it to behave in unintended ways, or leads to quality, cost, schedule, or performance shortfalls. Defects may be documented in problem reports (or trouble tickets), or they may be added to the backlog for consideration in future iterations. • **Escaped Defects** are defects detected, or resolved, after release of the product and version containing the defect. Defects are generally tracked separately for internal and external releases • **Contained Defects,** also known as Saves, are defects detected and resolved before internal or external release of the product and version containing the defect.
Change		Revision that adds, removes, or modifies any aspect of the product. Note: Identified changes may be documented using Stories or Features.
Change Request	CR	Requested change to the product. Some organizations may use Problem Reports instead of separate Change Requests to track issues.
Release	Build, Increment	A grouping of Capabilities and/or Features that can be used for demonstration, evaluation, or delivery. A release may be internal for integration, testing, or demonstration; or external, to system test or as user delivery. A release may be based on a time block or on product maturity.
Internal release		A release that is ready for internal use outside of the development team. It may be used for integration, testing, or demonstration.
Candidate Release	External Release	A release that has been through the pipeline and system test, and is ready for transition to the user.
Operational Release	Deployment Release	A release that has been approved for operational use.
Iteration	Sprint	A small internal time block in which the development team develops and demonstrates a set of Stories. An iteration is a full development cycle that can result in a Release. In some methodologies, an iteration is called a Sprint.

Term	Synonyms	Description
MVP / MVCR / NVP		**Minimum Viable Product (MVP):** An early version of the software to deliver or field basic capabilities to users for evaluation and feedback. Insights from MVPs help shape scope, requirements, and design of future product releases. **Minimum Viable Capability Release (MVCR)**: A set of features suitable to be fielded to an operational environment that provides value and capability to the end user in a rapid timeline. The MVCR delivers initial user capabilities to enhance some mission outcome(s). The MVCR, used in DOD software policy, is analogous to a Minimum Marketable Product (MMP) in commercial industry. **Next Viable Product (NVP)**: The next set of features in the succeeding product delivery.
Release Style		There are three types of release styles: Cadenced (e.g., Quarterly), Feature-based (e.g., Minimum Viable Product), and Continuous Deployment. Continuous Deployment takes significant discipline, and therefore requires more maturity. Most programs will do some form of cadenced release/iteration schedule, with specific time blocks.
Stakeholder		Individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations *(ISO/IEC/IEEE 15288:2015 Systems and software engineering--System life cycle processes),* *Examples: End users, end user organizations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, supplier organizations and regulatory bodies.*
Product		A product is the output of an enterprise that can be produced. There are four generic product categories: hardware (e.g., engine mechanical part); software (e.g., computer program); services (e.g., transport); and processed materials (e.g., lubricant).
Product Value		Product Value is an assessment of the degree to which the delivered product, capability, or service satisfies, or will satisfy the needs of its stakeholders including but not limited to mission improvements, efficiencies, risk reduction, and cost.

4. Mapping Data to Measurement Specifications

In the PSM methodology, the information model links the data that can be measured to a specified information need, as illustrated in Figure 5. More detail on the discussions in this section can be found in Practical Software and Systems Measurement (John McGarry (Author), 2001).

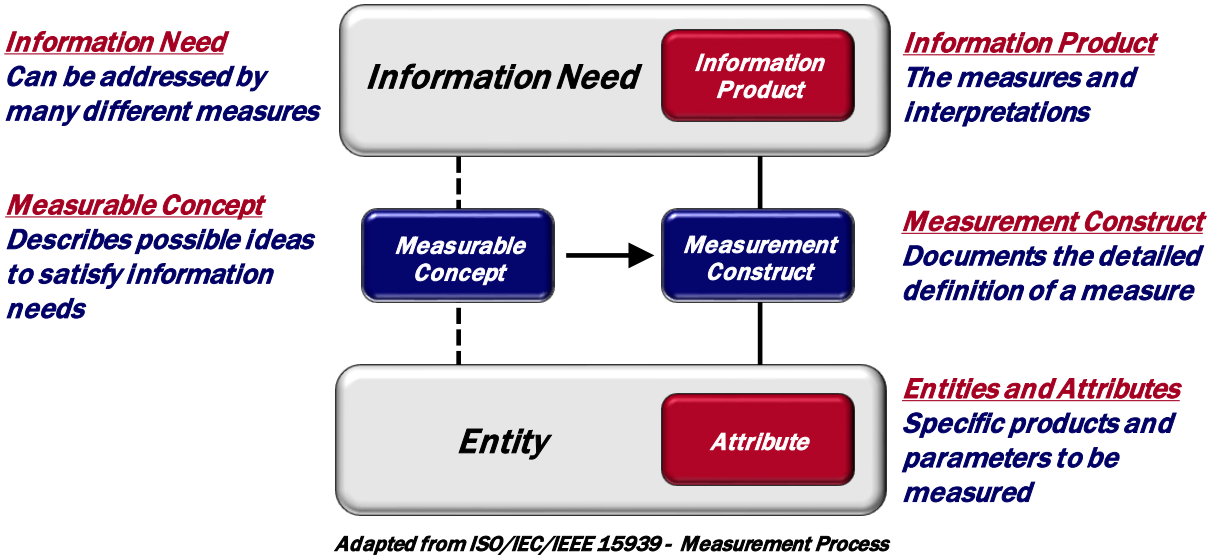


Figure 5: Information Model - High-Level View

The things that can actually be measured include specific attributes of the systems and software processes and products, such as size, effort, and number of defects. The measurement construct describes how the relevant attributes are quantified and converted to indicators that provide a basis for decision making. A single measurement construct may involve three types, or levels, of measures; base measures, derived measures, and indicators. The measurement planner needs to specify the details of the measurement constructs to be used in the measurement plan, as well as the procedures for data collection, analysis, and reporting.

At each of the three levels of measures - base measures, derived measures, and indicators - additional information content is added in the form of rules, models, and decision criteria. Figure 6 illustrates the structure of a measurement construct in more detail. This figure depicts how the base measures collected are dependent on the information needed by management. It also shows how the data is combined into an indicator and analysis model to form the information product provided to management.

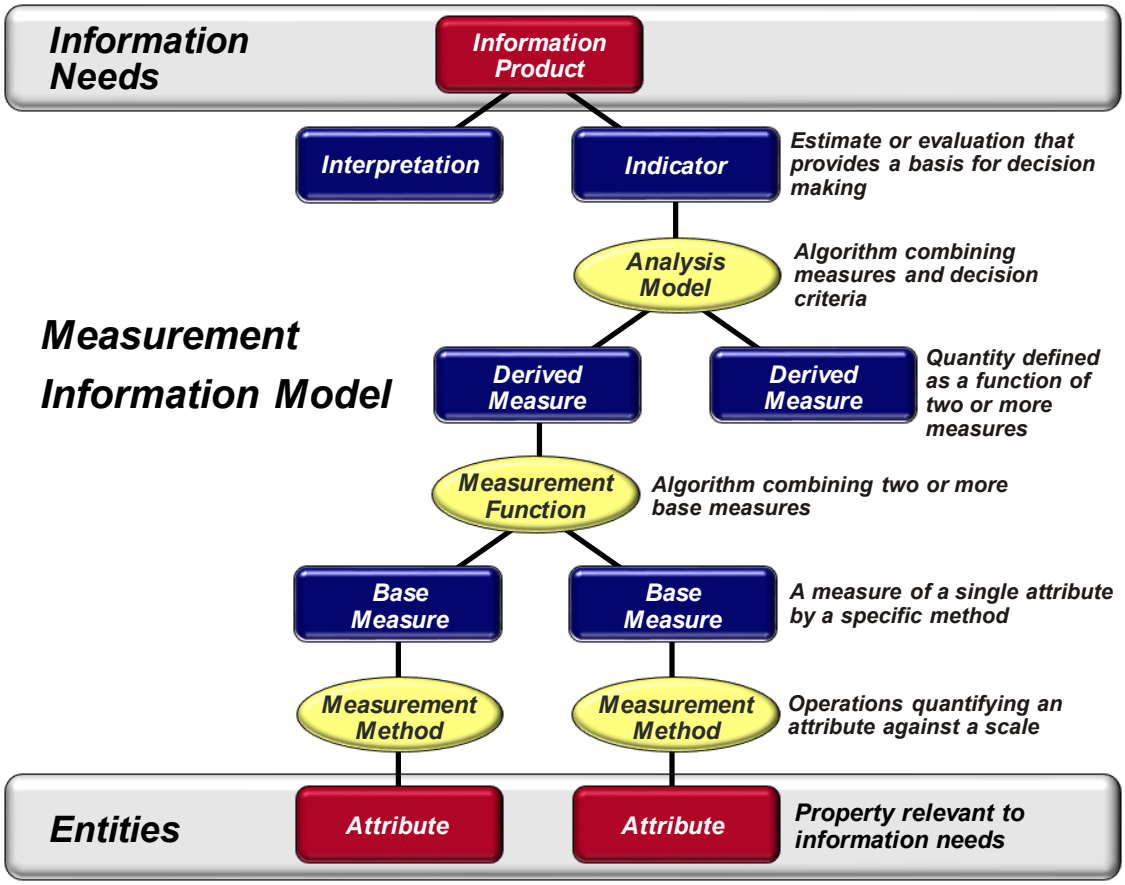


Figure 6: Measurement Information Model

Figure 7 contains a specific example of this, for the defect detection measure that is specified in Part 2, Section 8. The measurement specifications in Section 8 detail the information needs, base measures, derived measures, and analysis models for each proposed measure.

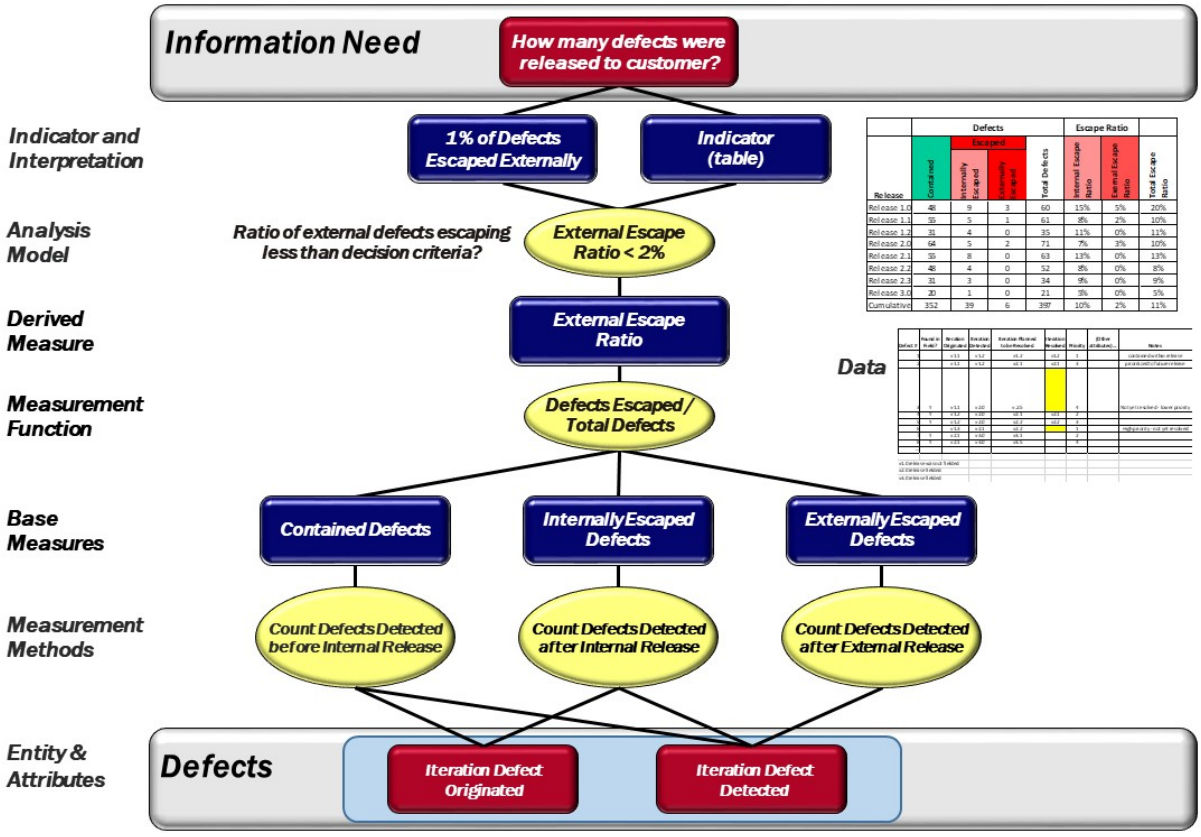


Figure 7: Mapping Data to Measures

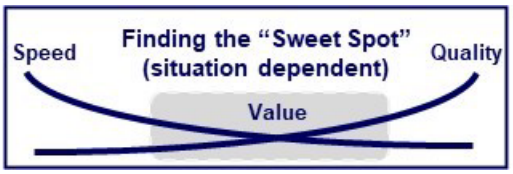
5. Measurement Principles

The "Information Category-Measurable Concept-Measures" (ICM) Table provides the PSM CID measurement framework detailing common information needs and measures that are effective for CID approaches. The information needs address team, product, and enterprise perspectives. These different perspectives have different information needs and concerns. In some cases, the same base measures may be aggregated to address high-level information needs. In other cases, unique measures are required. The ICM Table also identifies a set of measures that have been identified as being practical measures to address these information needs, based on practical experience from the working group members. The ICM table is included in Section 7.

Some key principles for these information needs and measures include:

- The set of measures included in the ICM Table are sample measures identified through survey and subject matter expert (SME) review as being important in selected circumstances and at various levels.
- Team, product, and enterprise measures are included: not all can be aggregated.
- A minimum practical set of measures should be selected and tailored based on organizational and program circumstances, tools, and processes. Often organizations or programs will select a subset of these measures to emphasize for implementation and decision-making.
- The selected measures should have an identified stakeholder, inform decisions or answer key programmatic questions, and drive actions. They allow early visibility into the issues so that timely corrective action can be taken.
- The set of measures are process agnostic, but they were specifically developed for continuous iterative development. Other PSM materials represent a broader set of materials and processes.
- The collection of measures should be automated to the extent practical and integrated with business workflows.
- A balance between speed and quality needs to be maintained, as illustrated in Figure 8. There is often a 'sweet spot' tradeoff between speed and quality that delivers a best value solution based on project objectives. Quality needs to be monitored, in addition to speed, to ensure that these measures are appropriately balanced. An over-emphasis on speed can be at the expense of product quality. An over-emphasis on quality can slow the speed of delivery. Some improvements (such as automation), can positively impact both speed and quality, and shift both curves.

Figure 8: Speed - Quality Sweet Spot



For the highest priority measures, sample measurement specifications have been developed that detail the identified measures. Measurement specifications have been developed for:

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

See Part 2, Section 8 for these specifications. The ICM table and the sample measurement specifications can also be found at http://www.psmsc.com/CIDMeasurement.asp.

Part 2, Section 9 also contains guidance and examples of the use of these high priority measures for enterprise decision making.

In addition, guidance on common information needs and associated measures for software assurance and technical debt have been developed. These are available in Part 3, sections 10 and 11.

6. Next Steps

This version of the PSM CID measurement framework is a set of measures that have proven to be useful in practice. Additional measures will be considered and added in future releases.

Potential future work areas include:

- Draft measurement specifications for software assurance and technical debt
- Size measures
- Estimation and Cost Prediction
  - Software effort size measures/drivers and uncertainty in estimating
  - How to quantify and assess early program estimates
  - Maintaining the cost baseline
- Additional measures for CID
  - Sprint Stability
  - Product Backlog Volatility (impact of scope changes)
  - For Defect Containment, consider removing Sprint/Iteration Containment and just address Release Containment.
  - Safety as a Quality Characteristic
- Update Product Value Measure
  - Based on feedback from user pilots
- Update all measures based on feedback from usage

7. ICM TABLE

Table 5: Issues, Categories, and Measures

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
1	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 capabilities/ requirements assigned 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
2	Schedule and Progress	Work Unit Progress		Did we deliver expected capabilities / features? Is the roadmap still valid?	Is the user satisfied with the delivered products? Does the system provide the desired functionality when needed?	Feature or Capability Implementation Product Value
3	Schedule and Progress	Work Unit Progress		Is integration and test progress proceeding as planned?		Test Progress
4	Schedule and Progress	Work Unit Progress		Is the flow of work moving forward through the process workflow states?		Cumulative Flow
5	Schedule and Progress	Work Backlog	How much outstanding technical or mission debt exists?	How much outstanding technical or mission debt exists?		Feature or Capability Backlog Burndown of Technical Debt Backlog Items

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
6	Schedule and Progress	Work Unit Progress Security	Are patches delivered as committed?	Are vulnerabilities and weaknesses resolved as committed?	What features/capabilities remain vulnerable and are unresolved?	Patches Delivered Vulnerabilities, Weaknesses Resolved Features/Capabilities Resolved Burndown of Vulnerabilities, Weaknesses
7	Resources and Cost	Financial Performance		What is the cost to release? (capability development through deployment)	What is the cost to release? (capability development through deployment)	Cost ($) Effort
8	Resources and Cost	Financial Performance		What is the estimated cost and schedule for a new CID product or release? What is the estimated cost and schedule per feature or capability?	What is the estimated cost and schedule for a reference feature or capability? (historical reference)	Estimate vs. Actual Cost/Effort Estimate vs. Actual Effort Estimate vs. Actual Schedule Earned Value
9	Resources and Cost	Financial Performance	Are the feature level estimates accurate and feasible?	Are the feature level estimates accurate and feasible?	How accurate are the estimates across the set of enterprise programs?	Committed vs. Completed Estimation Accuracy
10	Resources and Cost	Personnel Effort	Do we have the appropriate team members for each identified role (skills and skill levels) with appropriate training?			Staff Experience
11	Resources and Cost	Personnel Effort		How much turnover is occurring on the teams and as a whole?	How much turnover is occurring on the programs?	Team Turnover Rates Program Turnover Rates
12	Resources and Cost	Personnel Effort	What is the satisfaction of the workforce?	What is the satisfaction of the workforce?	What is the satisfaction of the workforce?	Net Promoter Score (NPS)
13	Resources and Cost	Facilities and Support Resources			How quickly can a new tool chain or environment be deployed?	Time to Deploy

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
14	Size and Stability	Functional Size and Stability Physical Size and Stability	How much work must be done?	How much work must be done?	How much work must be done?	Committed vs. Completed Requirements SLOC
15	Size and Stability	Functional Size and Stability		How volatile are capabilities or features? Are we adding more features? What is the ability to accommodate changes in user needs?	How volatile are capabilities or requirements? What is the ability to accommodate changes in user needs?	Feature Volatility Capability Volatility Backlog Volatility
16	Size and Stability	Functional Size and Stability	How much of the product is newly developed vs. reused from other sources?			Reuse of Artifacts
17	Size and Stability	Functional Size and Stability		What value is being provided?	What value is being provided?	Product Value Mission Effectiveness Business Value
18	Product Quality	Functional Correctness	Do features/stories work as expected?	Do features/capabilities work as expected?	Do capabilities work as expected? Is rework identified and managed?	Acceptance of Completed Work (Stories, Features, Capabilities) Rework Stories Enhancement Stories Defect Detection Defect Resolution
19	Product Quality	Functional Correctness	Do changes break previous functionality?	Do changes break previous functionality?	Do changes break previous functionality?	Rework Defects Rework Hours Rework Stories Change Failure Rate Rollback Defect Density

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
20	Product Quality	Functional Correctness	How many defects were contained (discovered) prior to internal release? How many defects were released (escaped) to an internal customer (e.g., Integration and Test, Formal Test)?	How many defects were released (escaped) to an internal customer (e.g., Integration and Test, Formal Test) or released (escaped) to an external customer (e.g., end users)?	How many defects were released (escaped) to an external customer (e.g., end users)?	Defect Detection
21	Product Quality	Functional Correctness	What is the product quality delivered from the development team?	What is the product quality delivered to the field?	What is the product quality delivered to the field?	Defect Detection Defect Resolution Defect Density
22	Product Quality	Value	Do features/stories work as expected?	Does the delivered product meet the operational need?	Does the delivered product meet the mission need?	Product Value
23	Product Quality	Functional Correctness Security	How many software assurance defects have been identified and adjudicated?	How many software assurance defects have been identified and adjudicated? How big/what is the size of the system's attack surface? Is the attack surface increasing, decreasing, or staying the same?	How big/what is the size of the system's attack surface? Is the attack surface increasing, decreasing, or staying the same?	Common Vulnerabilities Enumeration (CVEs) Common Weaknesses Exposure (CWEs) CVEs/CWEs Detected / Resolved Software Assurance Defects Detected / Resolved Size of Attack Surface
24	Product Quality	Supportability – Maintainability Dependability - Reliability		What is the reliability and availability of operational capabilities? How long does it generally take to restore service when a service incident occurs (e.g., unplanned outage, service impairment)?	What is the reliability and availability of operational capabilities? How long does it generally take to restore service when a service incident occurs (e.g., unplanned outage, service impairment)?	Mean Time to Restore (MTTR) Mean Time to Detect (MTTD)

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
25	Product Quality	Supportability – Maintainability Dependability - Reliability	What is the reliability and availability of the environment (e.g., people, process, infrastructure)?	What is the reliability and availability of the environment (e.g., people, process, infrastructure)?	What is the reliability and availability of the environment (e.g., people, process, infrastructure)?	Environment Reliability
26	Product Quality	Security	How many vulnerabilities and weaknesses were inherited from COTS? How many have been mitigated? How many have been reported to the National Vulnerability Database (NVD)? ?	How many vulnerabilities and weaknesses were inherited from COTS? How many have been mitigated? How many have been reported to the National Vulnerability Database (NVD)? What percentage of code from suppliers (legacy, 3rd party, subcontractors, COTS) is screened for vulnerabilities?	What is the quality / vulnerability / supportability of legacy and third party code?	Percentage of Code Base Available for Screening Percentage of Code Base Screened for Vulnerabilities Percentage of Code Requiring Binary Analysis (no source code available)
27	Process Performance (Process Effectiveness)	Safety		Have all safety-critical items been resolved?	Is the system safe to operate?	Safety Assessment Status
28	Process Performance	Process Efficiency – Speed Security		How long does it take to successfully complete software assurance audit/penetration testing?	How long does it take to successfully complete software assurance audit/penetration testing?	Software Assurance Test Duration

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
29	Process Performance	Process Efficiency Security	How often has the baseline changed? Is the baseline stable?	How long does it take to get an Authority to Operate (IATT/ATO) approval for new releases? How long does it take to prepare the authorization Package? Is the Time to Authorization quick enough to meet the criteria of a Continuous ATO? How many critical software assurance defects are holding up/present a roadblock to the authorization process?	How long does it take to get an Authority to Operate (IATT/ATO) approval for new releases? How fast can the system deploy new secure capabilities to users? Can we release the system (Go/No Go Decision)?	Time to authorization (IATT/ATO) Time to Prepare the authorization Package Authorization (IATT/ATO) Status Frequency of Baseline Changes Unresolved Critical Software Assurance Defects
30	Process Performance	Process Efficiency - Speed	Is the flow of work (stories) moving forward through the value stream? Is the flow of work as efficient and predictable as needed?	Is the flow of work (features, capabilities) moving forward through the value stream? Is the flow of work as efficient and predictable as needed?	Are the evolving stakeholder needs being met when needed?	Committed vs. Completed Cumulative Flow Capacity
31	Process Performance	Process Efficiency - Speed	Is the team performing as expected? How much work can be accomplished by a team in a future iteration?	n/a	n/a	Team Velocity Acceleration
32	Process Performance	Process Efficiency - Speed		How long does it take to deploy an identified feature/capability?	How responsive is the program to change?	Cycle Time / Lead Time Release Frequency

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
33	Process Performance	Process Efficiency - Speed		What is the cadence of product release or deployment? How long does it take to release a minimum viable product?	What is the cadence of product release or deployment? How long does it take to release a minimum viable product?	Release Frequency MVP Release Duration
34	Process Performance	Process Efficiency - Speed		How much time does it take to conduct a full regression test? How much time for the automated regression test?		Test Duration Automated Test Duration
35	Process Performance	Process Effectiveness		How much of the testing is automated? How often do we perform automated testing? How much capability is tested in an automated fashion?	How much of the system testing is automated? How much of user test is automated? How often do we perform automated testing? How much of system automated test is credited for user test?	Automated Test Coverage Automated Test Frequency
36	Process Performance	Process Effectiveness - Value		What is the product value (normalized feature / capability delivered by effort)? Is productivity improving over time?	What is the product value (normalized feature / capability delivered by effort)? Is productivity improving over time?	Product Value Team Velocity Acceleration
37	Process Performance	Process Effectiveness	Is the work in progress being managed appropriately?	Is the work in progress and product backlog being managed appropriately? Are there queues or delays in our process workflows that prevent us from optimizing throughput?	Are there (major) queues or delays in our process workflows that prevent us from optimizing throughput?	Cumulative Flow Defect Resolution Backlog Readiness

Row	Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
38	Customer Satisfaction	Customer Support			Is the user satisfied with the delivered products? Does the system provide the desired functionality when needed?	Product Value

BIBLIOGRAPHY

Policy and Study Board Reports

Software Acquisition Pathway Interim Policy and Procedures. (2020, January 3). Retrieved from Defense Acquisition University: https://aaf.dau.edu/

Defense Innovation Board (DIB), *Software Is Never Done: Refactoring the Acquisition Code for Competitive Advantage*, 2019, Software Acquisition and Practices (SWAP)

Defense Science Board (DSB), *Design and Acquisition of Software for Defense Systems*, Defense Science Board (DSB) Task Force on Design and Acquisition of Software for Defense Systems, 2018

https://dsb.cto.mil/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf

General

Design and Acquisition of Software for Defense Systems, Defense Science Board (DSB) Task Force on Design and Acquisition of Software for Defense Systems. (2018, February). Retrieved from Defense Science Board: https://dsb.cto.mil/reports/2010s/DSB_SWA_Report_FINALdelivered2-21-2018.pdf

John McGarry (Author), D. C. (2001). *Practical Software Measurement: Objective Information for Decision Makers.* Addison-Wesley Professional.

Software Acquisition Pathway Interim Policy and Procedures. (2020, January 3). Retrieved from Defense Acquisition University: https://aaf.dau.edu/

Software Is Never Done: Refactoring the Acquisition Code for Competitive Advantage. (2019, May 3). Retrieved from https://media.defense.gov/2019/Apr/30/2002124828/-1/-1/0/SOFTWAREISNEVERDONE_REFACTORINGTHEACQUISITIONCODEFORCOMPETITIVEADVANTAGE_FINAL.SWAP.REPORT.PDF

Vacanti, D. S. (2015). *Actionable Agile Metrics for Predictability: An Introduction.* Daniel S. Vacanti, Inc.