Digital Engineering Measurement Framework – v1.1

A collaboration among industry, government and academia

Developed and Published by Members of:

Practical Software & Systems Measurement

Systems Engineering Research Center

Aerospace Industries Association

National Defense Industrial Association

International Council on Systems Engineering

Department of Defense Research & Engineering

The Aerospace Corporation

- DISTRIBUTION A.
- Approved for public release: distribution unlimited.

Digital Engineering Measurement Framework - Project Overview and Timeline

2020

AIA EMC Project Plan

- Refined list of DE metrics serving as Key Performance Indicators for program execution, and model health
- Detailed descriptions of each metric, traceable to SE metrics, quality, & requirements volatility

Established collaborative WG (9/14/20)

(PSM, NDIA, INCOSE, AIA, SERC, Aerospace, OUSD R&E, …)

Objectives

- Define industry consensus measurement framework for DE, MBSE
- Align measures with business information needs for project execution and organizational performance improvement.

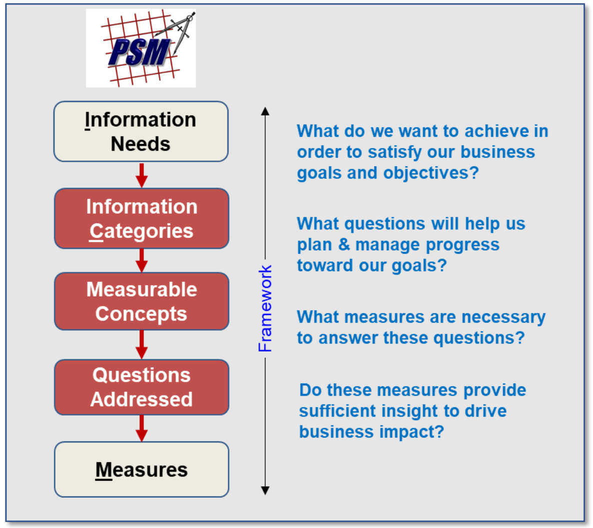
Leverage partner resources and assets

- Practical Software and Systems Measurement (PSM) [Continuous Iterative Development Measurement Framework](#)
- [SERC / INCOSE / NDIA MBSE Maturity Survey](#)
- SERC DE metrics research ([SERC-2020-SR-003](#), [SERC-2020-TR-002](#))
- [Systems Engineering Leading Indicators Guide](#)
- [DoD Digital Engineering Strategy](#)

2021

Follow PSM process to define DE measurement framework

- Aligned with ISO/IEC/IEEE 15939 measurement process standard

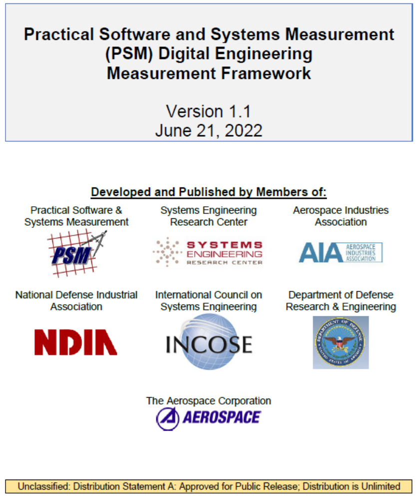


Team product development

- Front matter (concepts, terms, …)
- Information Needs (ICM Table)
- Measurement specifications

2022

Initial framework draft for review (Jan 2022) Publication release (June 2022)


Practical Software and Systems Measurement (PSM) Digital Engineering Measurement Framework

Version 1.1
June 21, 2022

Developed and Published by Members of:

Practical Software & Systems Measurement | Systems Engineering Research Center | Aerospace Industries Association

National Defense Industrial Association | International Council on Systems Engineering | Department of Defense Research & Engineering

The Aerospace Corporation

Unclassified: Distribution Statement A: Approved for Public Release; Distribution is Unlimited

Initial Measurement Specifications

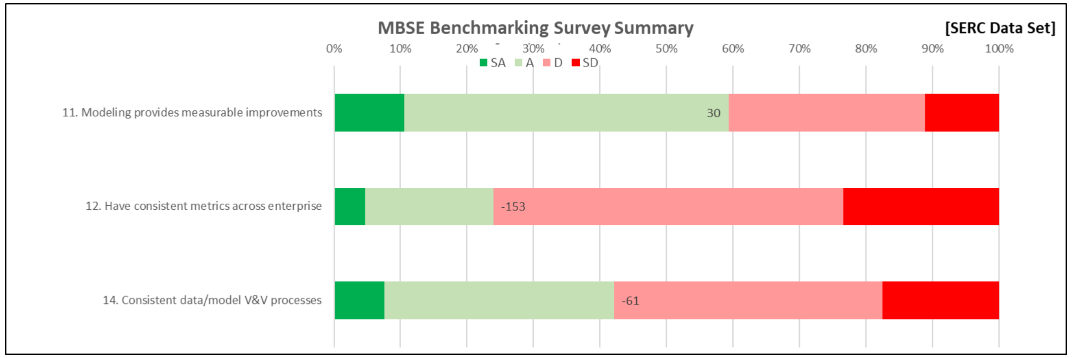
- Architecture Completeness and Volatility
- Model Traceability
- Product Size
- DE Anomalies
- Adaptability and Rework
- Product Automation
- Deployment Lead Time
- Runtime Performance

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

Lack of effective DE/MBSE measures has been an inhibitor to digital transformation

Substantiated by DoD SERC research

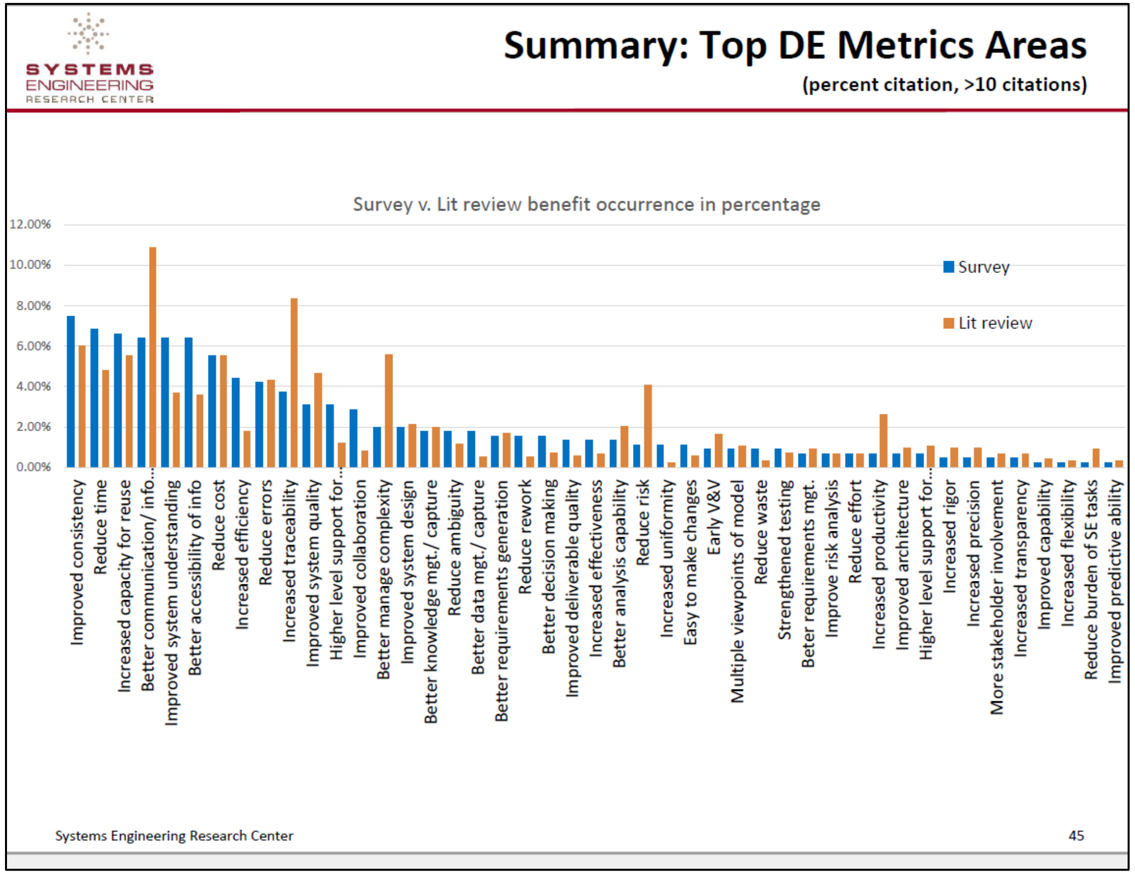
Benchmarking the Benefits and Current Maturity of Model-Based Systems Engineering across the Enterprise (SERC-2020-SR-001)



Category	Question title	SERC MBSE Questionnaire	Survey Scor	SA	A	D	SD	Chart	Calculated Scc
Model Metrics	11. Modeling provides measurable improvements	Modeling activities in our organization provide measurable improvements within and across projects.	30	18	83	50	19		30
	12. Have consistent metrics across enterprise	We have consistent metrics across our program(s)/enterprise that include our modeling activities.	-153	8	33	90	40		-153

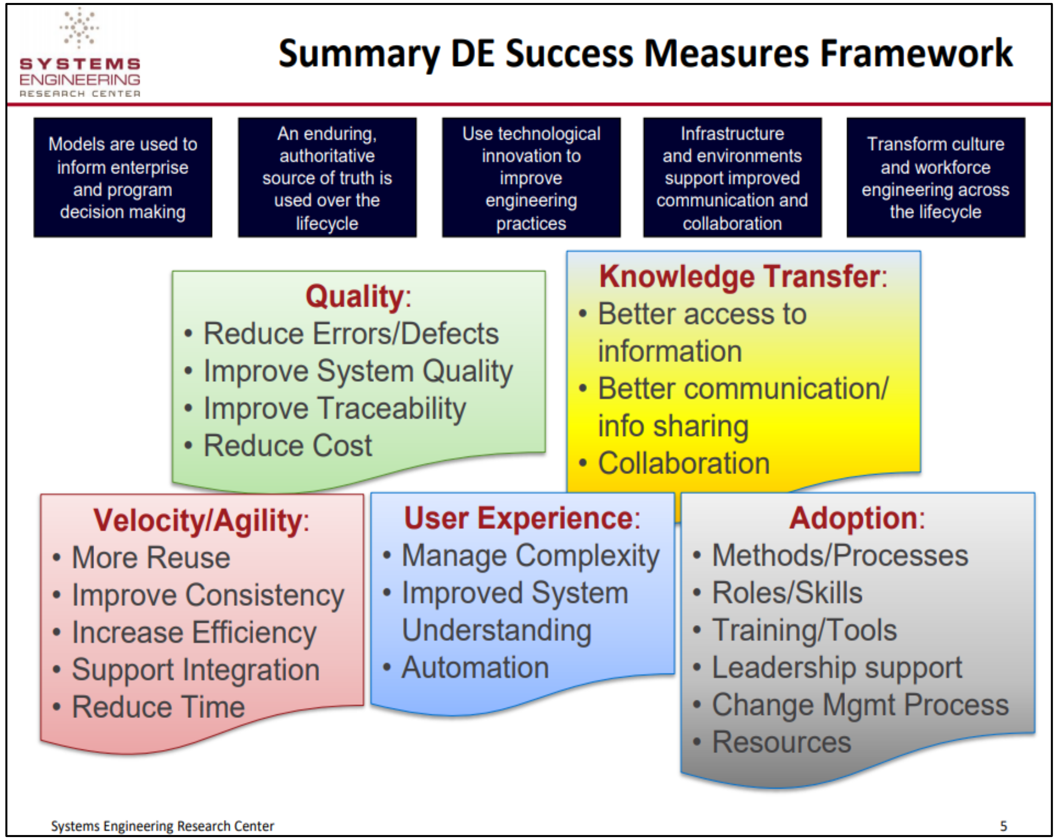
https://sercuarc.org/results-of-the-serc-incose-ndia-mbse-maturity-survey-are-in/

Summary Report Task Order WRT-1001: Digital Engineering Metrics Supporting Technical Report (SERC-2020-SR-003)
Task Order WRT-1001: Digital Engineering Metrics Technical Report (SERC-2020-TR-002)



Success Measures and Benefits of Digital Engineering Transformation

Research from DoD SERC and Virginia Tech helped inform the DE Measurement Framework



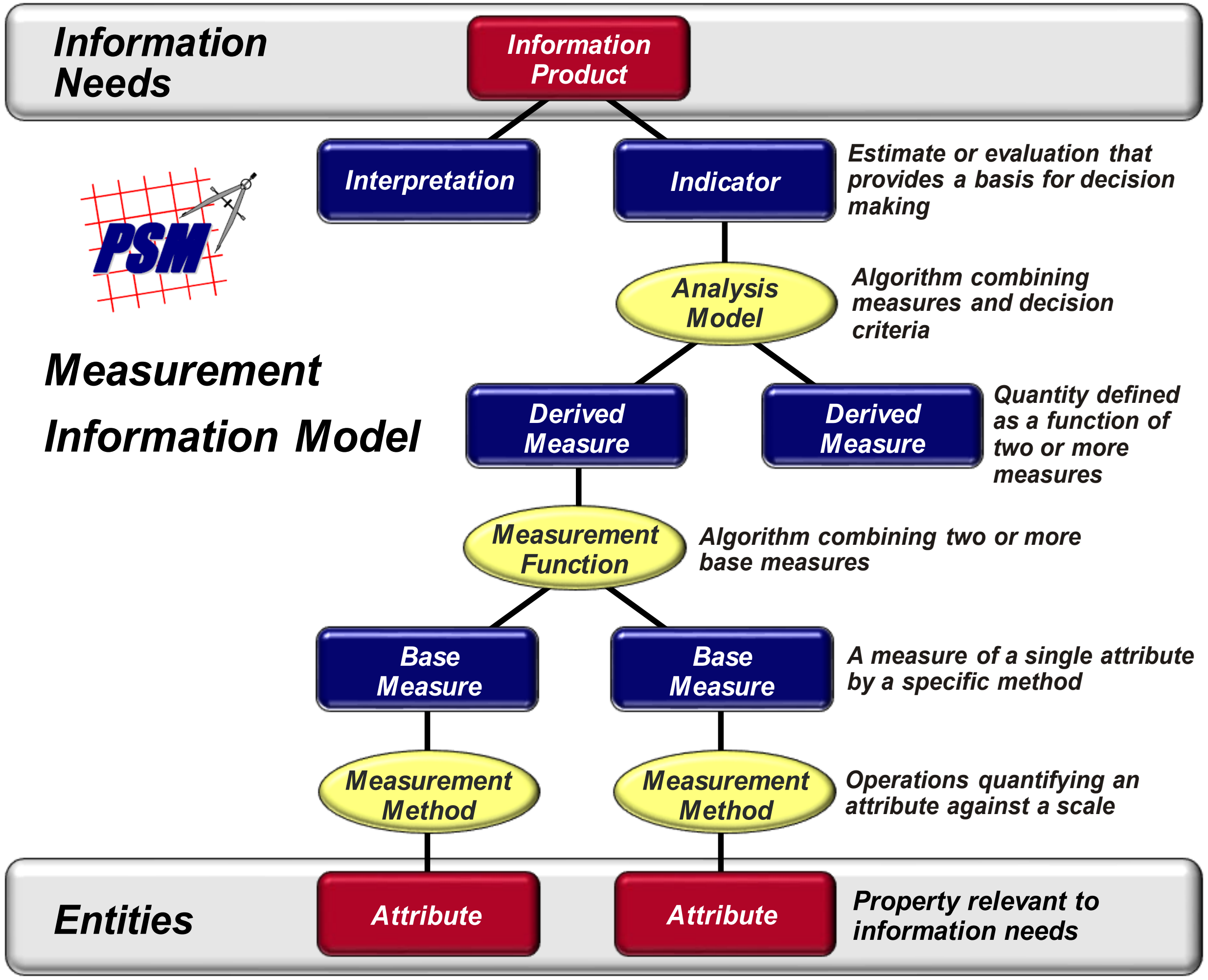
Primary Benefits	Description	Applicable Measurement Specifications
Higher level support for automation	Use of tools and methods that automate previously manual tasks and decisions	8.6 Product Automation 8.7 Deployment Lead Time
Early Verification and Validation (V&V)	Moving tasks into earlier developmental phases that would have required effort in later phases	8.4 DE Anomalies 8.5 Adaptability and Rework 8.7 Deployment Lead Time
Reusability	Reusing existing data, models, and knowledge in new development	8.4 DE Anomalies 8.5 Adaptability and Rework 8.7 Deployment Lead Time
Increased Traceability	Formally linking requirements, design, test, etc. via models	8.7 Deployment Lead Time 8.8 Runtime Performance
Strengthened Testing	Using data and models to increase test coverage in any phase	8.1 Architecture Completeness and Volatility 8.2 Model Traceability 8.3 Product Size
Better Accessibility of Information (ASoT)	Leveraging an Authoritative Source of Truth (ASoT) to increase access to digital data and models to increase the involvement of stakeholders in program decisions	8.7 Deployment Lead Time 8.8 Runtime Performance
Higher Level of Support for Integration	Using data and models to support integration of information and to support system integration tasks	8.6 Product Automation 8.2 Model Traceability
Multiple Model Viewpoints	Presentation of data and models in the language and context of those that need access	8.1 Architecture Completeness and Volatility 8.7 Deployment Lead Time

PSM measures are derived from business information needs

Based on objectives and issues from the project or 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

Measures should provide insight into project or enterprise information needs to support decision-making



PSM Practical Software and Systems Measurement, www.psmsc.com

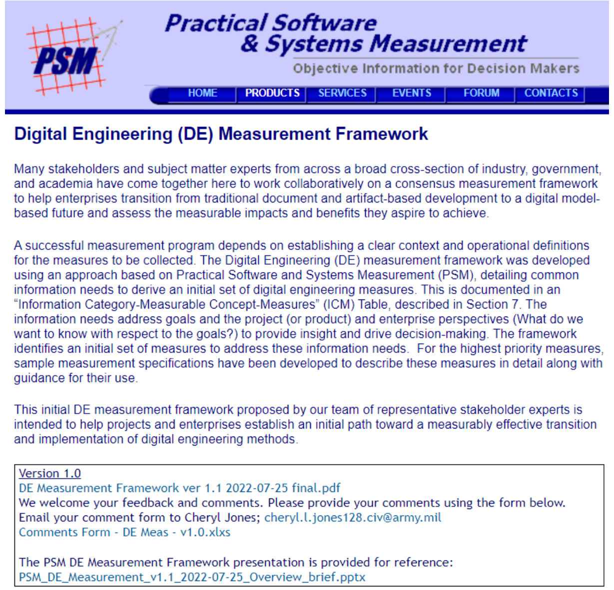
See Framework for more information

DE Measurement Framework ICM Table (Excerpt)

* = Measurement specs written for inclusion in v1.1 release

Information Categories	Measurable Concepts	Project Information Needs	Enterprise Information Needs	Potential Measures	Notes (Guiding Objectives)
Product Quality	Functional Correctness	Are we finding and removing anomalies early in the life cycle using models and shared information? Is the quality of the product in question adequate for the product to be used in subsequent phases or activities?	How many anomalies were released (escaped) to operations? Is the use of DE leading to the detection of anomalies earlier in the lifecycle compared to traditional methods or projects)? Has the detection curve shifted to the left?	DE Anomalies *	For digital engineering focus on the defects for modeling and simulation (including drawings).
Product Quality	Functional Correctness	How much rework effort is spent maintaining planned or unplanned changes to DE work products across the life cycle?	How much is rework reduced through use of DE? Can changes to work products be implemented more efficiently and with less effort in a DE environment relative to traditional methods?	Adaptability and Rework * Acceptance of Completed Work Products (Model Elements, Artifacts) Rework or Rework Defects	Completion of work products requires defined acceptance criteria. Rework is required when the acceptance criteria are not met.
Product Quality	Functional Correctness	What traceability gaps or defects exist in the digital model? Does model traceability support change impact assessments (requirements, design, compliance)?	Is architectural traceability improved using digital engineering methods relative to traditional approaches?	Model Traceability * Traceability Anomalies	
Process Performance	Process Effectiveness	How many released, validated system definitions/analyzed elements were functionally correct, but returned for rework?	Is the organization learning how to reduce the number of defects released to operations?	Model Element DE Anomalies	

Summary of v0.95 Public Review Comments


Practical Software & Systems Measurement
Objective Information for Decision Makers
HOME | PRODUCTS | SERVICES | EVENTS | FORUM | CONTACTS

Digital Engineering (DE) Measurement Framework

Many stakeholders and subject matter experts from across a broad cross-section of industry, government, and academia have come together here to work collaboratively on a consensus measurement framework to help enterprises transition from traditional document and artifact-based development to a digital model-based future and assess the measurable impacts and benefits they aspire to achieve.

A successful measurement program depends on establishing a clear context and operational definitions for the measures to be collected. The Digital Engineering (DE) measurement framework was developed using an approach based on Practical Software and Systems Measurement (PSM), detailing common information needs to derive an initial set of digital engineering measures. This is documented in an "Information Category-Measurable Concept-Measures" (ICM) Table, described in Section 7. The information needs address goals and the project (or product) and enterprise perspectives (What do we want to know with respect to the goals?) to provide insight and drive decision-making. The framework identifies an initial set of measures to address these information needs. For the highest priority measures, sample measurement specifications have been developed to describe these measures in detail along with guidance for their use.

This initial DE measurement framework proposed by our team of representative stakeholder experts is intended to help projects and enterprises establish an initial path toward a measurably effective transition and implementation of digital engineering methods.

Version 1.0
DE Measurement Framework ver 1.1 2022-07-25 final.pdf
We welcome your feedback and comments. Please provide your comments using the form below.
Email your comment form to Cheryl Jones; cheryl.l.jones128.civ@army.mil
Comments Form - DE Meas - v1.0.xlsx

The PSM DE Measurement Framework presentation is provided for reference:
PSM_DE_Measurement_v1.1_2022-07-25_Overview_brief.pptx

https://www.psmsc.com/DEMeasurement.asp

Technical comments are resolved.

Final editorial review in v1.1 release (consistency, cleanup)

Count of Resolution	Column L								
Organization	A	AIP	AIP / Defer	AIP / R	Defer	Defer/AIP	N/A	R	Grand Total
Aerospace	36	36	6	2	3		1	9	93
AFIT	1	1	1	1	3				7
Airbus	1	4			1			5	11
ALP International	1								1
Collins Aerospace	1	1			1				3
DAU		6	2		1	2			11
DHS	2	3	2		1			3	11
DoD		1			1				2
General Dynamics	4	7	2		1				14
GTRI	13	9	1						23
INCOSE	6	23	4		4		5	2	44
L3Harris	4	6						1	11
Lockheed Martin	6	12	2	2	3		1	6	32
MITRE	10	21			1			7	39
MxD			1						1
Naval Postgraduate School (NPS)		2			1			2	5
OSD DOT&E	5	15						8	28
OUSD R&E	18	12			2			5	37
PSM	17	8							25
Raytheon	14	13		1	2			7	37
SEI		3			1		2		6
U.S. Army	1	4							5
U.S. Marine Corps								4	4
Volkswagen	12	6	1						19
U.S. Navy		1							1
NRO	3	3							6
Grand Total	**155**	**197**	**22**	**6**	**26**	**2**	**9**	**59**	**476**

Significant changes from v0.95 to v1.0
- **Clarify terms and definitions**
- **Address editorial and technical comments to indicator specs**
- **Consolidate Defect Detection and Defect Resolution specs to 'DE Anomalies'**
- **Provide additional guidance in measurement specs and indicators**
- **Expanded descriptions of applicable life cycle models (agile, DevOps, waterfall…)**
- **Some document section restructuring for readability**

A	**Accept**
AIP	**Accept in Principle**
R	**Rejected**
Defer	**Deferred to a future release**
N/A	**Comment only, no action**

Of 476 public review comments received:
380 Accepted in full or in principle (80%)
26 Deferred to a future release (5%)
59 Rejected (12%)
Others in some partial combination of the above

183 Editorial (38%)
291 Technical (61%)
- **High: 131 (27%)**
- **Low: 160 (34%)**

Example Measurement Information Model – DE Anomalies

Changed term “Defects” to “Anomalies” based on review feedback , and consolidated 2 prior Defect Detection vs. Defect Removed indicators

Digital engineering measures and indicators are specified in a structured template aligned with the PSM Measurement Information Model

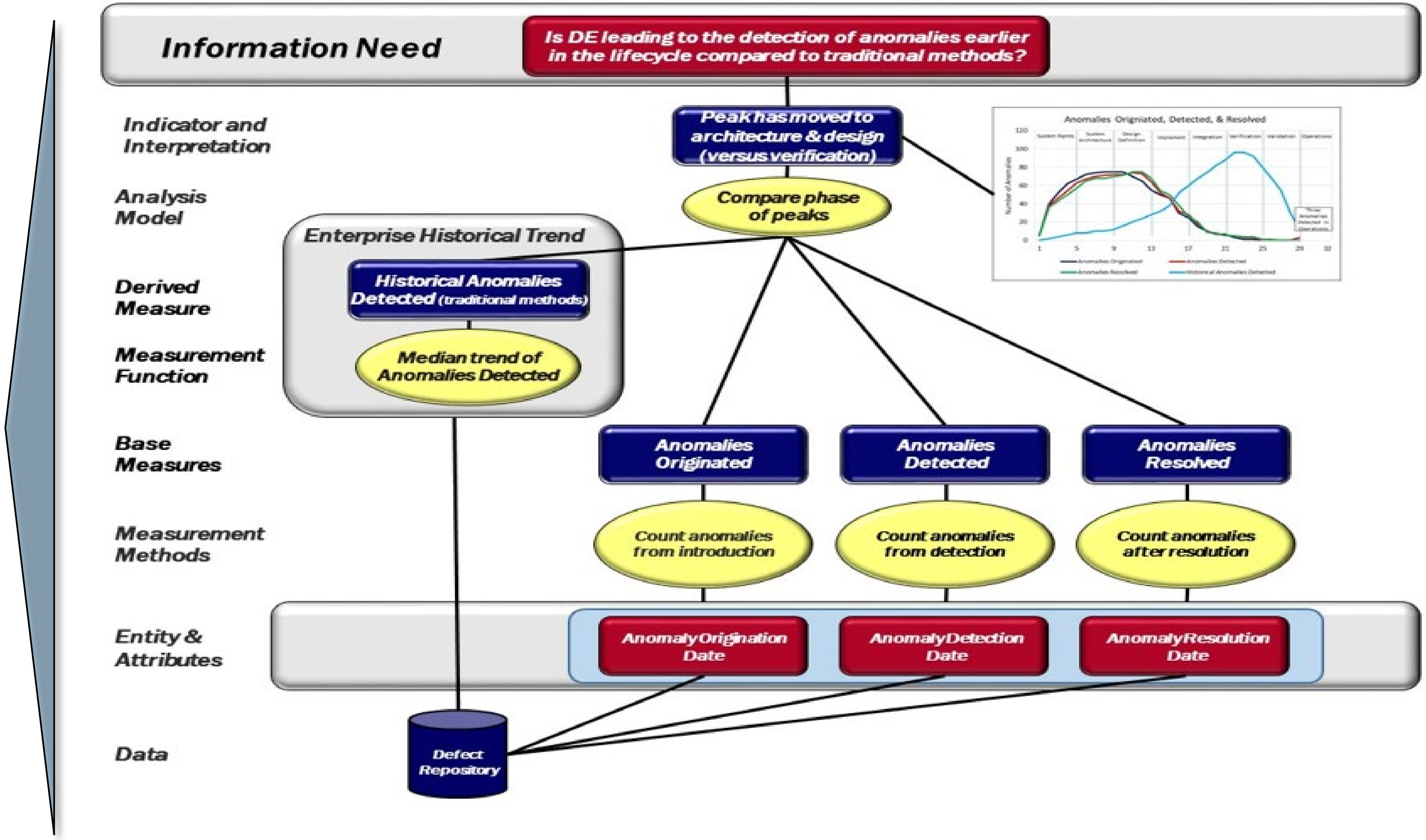


Figure 3.2-4: Mapping Data to Measures

Example Measurement Specification (Excerpts)

8.2 MODEL TRACEABILITY

Description

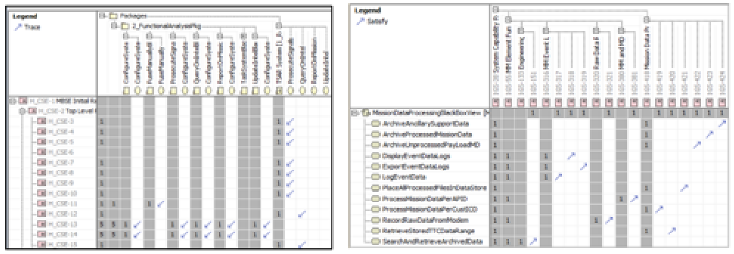
Measure Introduction		
Description	The usefulness and quality of a digital model depends on the completeness and integrity of the relationships among model elements. Traceability between elements, such as requirements allocation and flow down to architectural, design, and implementation components, assures that the system solution is complete and consistent. Gaps in bi-directional traceability between the artifacts of two models or might indicate where further analysis or refinement are needed. This might further apply to traceability gaps within a single model. when there is no implicit traceability between artifacts of different design stages. The prerequisites of any traceability measurement are agreed-upon, a priori guidelines and definitions, e.g., what model elements and relationships shall be traced, that apply to the specific DE model of the system. *Note:* While traceability might be applied to any model elements of interest that shall be defined a priori, functional architecture completeness always explicitly focuses on functions, requirements, and the associated hierarchy. Traceability reports and analyses might be facilitated by digital modeling tools. The traceability concepts and indicators in this specification are representative examples of more general traceability mappings and reports across the development life cycle, such as: • Traceability between stakeholder needs, system requirements, and allocated or derived requirements at each level of the system hierarchy • Traceability and flow down of requirements to the logical or physical solution domain (e.g., design, implementation, integration, verification, validation) • Allocation and traceability of performance measures or parameters, such as Measures of Effectiveness (MOEs) or Key Performance Parameters (KPPs) • Traceability of system interfaces	
Relevant Terminology	Model Element	Modeling constructs used to capture the structure, behavior, and relationships among system model components (See 2.2.2 Model Element)
	Source Element	The *a priori* base model elements defined per DE model from which other model elements shall be derived from or allocated to, e.g., a stakeholder needs.
	Destination Element	The model elements defined per DE model that shall be derived from or allocated to the Source Elements.
	Traceability Gap	One or more model elements defined per DE model that shall be traced, but that have not yet been derived or allocated to Source Elements.
		Note: For enhanced traceability concepts refer to the advanced topic discussion.

Definitions

Measures (Base, Derived)

Information Need and Measure Description	
Information Need	What is the extent of achieved traceability coverage from Source Elements, e.g., requirements, down to the logical or physical solution domain? What is our progress in completing the digital model? What traceability gaps exist?
Base Measure 1	Model Elements Traced [integer] "Number of model elements in a 1.. n source/destination element relationship(s) as defined in an agreed upon, *a priori* guideline.
Base Measure 2	Model Elements Not Traced [integer] Number of model elements not in any 1.. n source/destination *element relationship as* defined in an agreed upon, *a priori* guideline.
Derived Measure 1	Total Model Elements = Model Elements Traced + Model Elements Not Traced [integer] Total number of model elements *Note:* As defined in an agreed upon, a priori guideline (See Base Measure 1 and Base Measure 2).

Indicator(s) and Interpretation

Indicator Specification	
Indicator Description and Sample	Model Traceability can be depicted using visual or tabular summaries of the relationships among model elements. The specific indicators may depend on the model elements for which traceability is being measured, and the built-in reports and analyses provided by the digital modeling tool. For example, traceability among model elements might be implemented by showing requirements derivation and model traceability coverage of stakeholder needs into system and component requirements. Representative example indicators used to assess traceability dependencies among selectable model elements (e.g., requirements, use cases, activities, logical architecture and design, physical design, interfaces, parameters, measures of performance) are depicted in Figure 8.2-1. Here, mostly 2-dimensional matrices containing model specific model elements of interest are utilized. Alternatively, the relationship between model elements might be depicted as flow down. With respect to Figure 8.2-1 (bottom left), a specific use case is linked to related actions via an activity diagram.  Traceability Between Model Elements (Dependency Matrix) Relationships to Problem or Solution Domain («satisfy» or «refine» Matrix)

Analysis Guidance

Analysis Model	Projects and organizations shall define the objectives, constraints, and criteria for establishing traceability among applicable model elements. This is typically guided by a model schema, metamodel, or blueprint that constrains traceability to meet the model's purpose. Review and analyze traceability dependencies among model elements to assess the completeness, adequacy, quality, and integrity of the digital model. The analysis may vary according to the types of specific model elements selected, but general guidelines may include: • Each source (parent) model element (Model Element 1) should be traceable to one or more allocated or derived destination (child) model elements (Model Element 2). • Each destination (child) model element (Model Element 2) should be derived from, or refine, a parent requirement or model element (Model Element 1). • Determine if the set of linked dependencies are, in aggregate, sufficient to adequately implement the parent requirement or model element.
Decision Criteria	In case a desired model traceability coverage (Derived Measure 2), e.g., 70%, of model elements of interest has not been met, the team shall specifically address these gaps. To validate whether the system meets stakeholder needs, at minimum, the system requirements should be traceable to these stakeholder needs. Model elements that do not satisfy requirements, might be obsolete and shall be evaluated. Again, the prerequisites of any decision making are agreed-upon, a priori guidelines and definitions, e.g., what model elements and relationships shall be traced, that apply to the specific DE model of the system

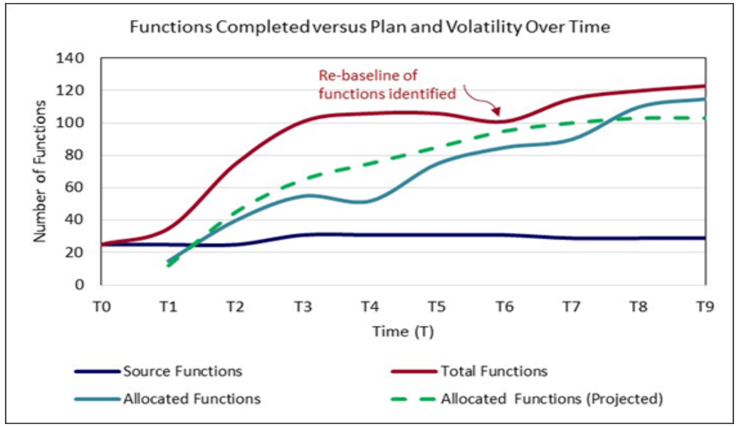
Additional Information and Guidance

Additional Analysis Guidance
Implementation Considerations

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

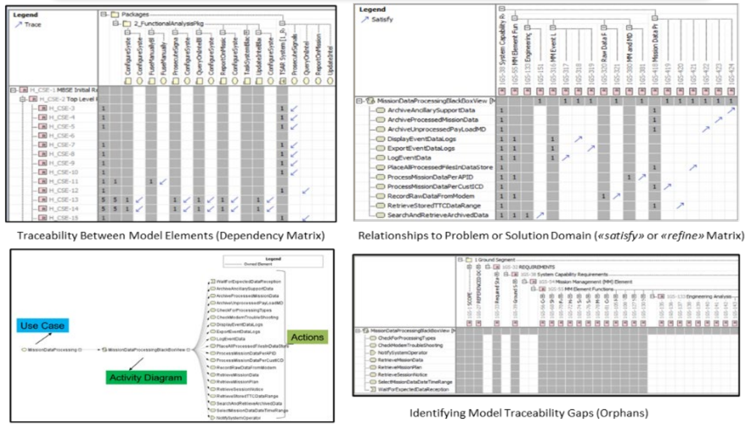
Digital Engineering Measurement Framework – Example Indicators

Architecture Completeness and Volatility



Is the architecture complete to proceed with design?

Model Traceability



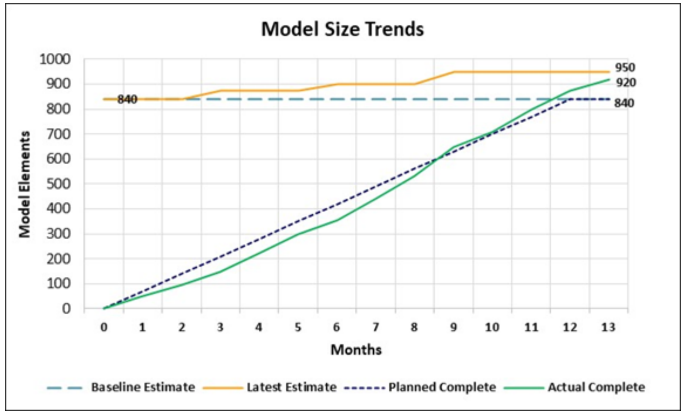
Traceability Between Model Elements (Dependency Matrix)

Relationships to Problem or Solution Domain (*«satisfy»* or *«refine»* Matrix)

Identifying Model Traceability Gaps (Orphans)

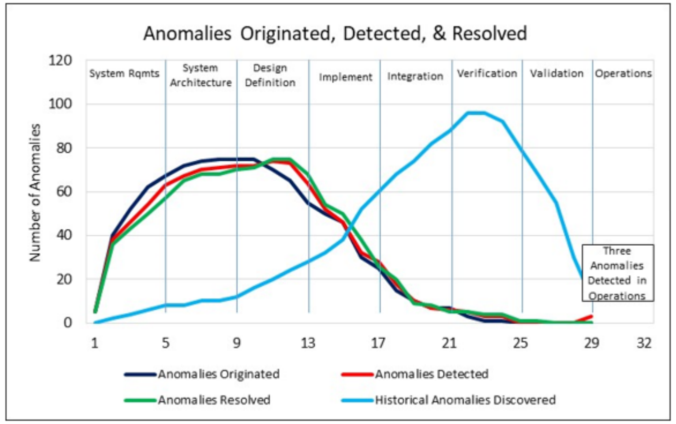
What is the traceability and coverage of model elements?

Product Size (Model Elements)

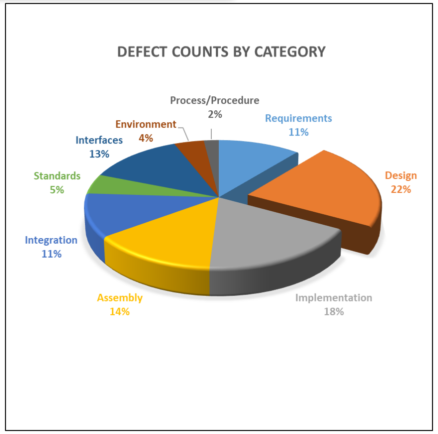


What is the size and scope for the DE project or product?

DE Anomalies



Are we finding and removing anomalies earlier using DE?



How can we reduce the leading causes of anomalies?

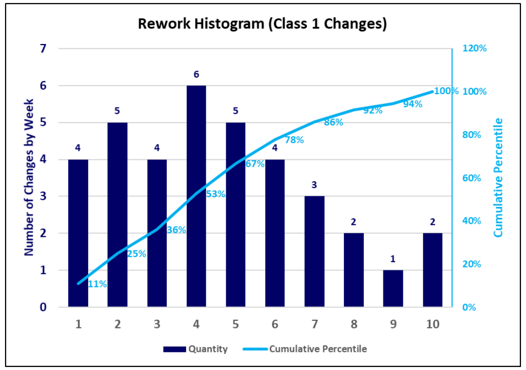
Excerpts only from DE measurement specifications. Some specs have multiple sample indicators. See framework Section 8 - Measurement Specifications for details.

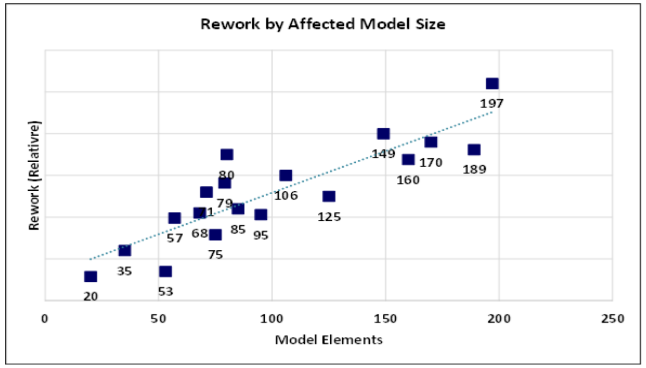
Click titles for details

Digital Engineering Measurement Framework – Example Indicators

Excerpts only from DE measurement specifications. Some specs have multiple sample indicators. See framework Section 8 - Measurement Specifications for details.

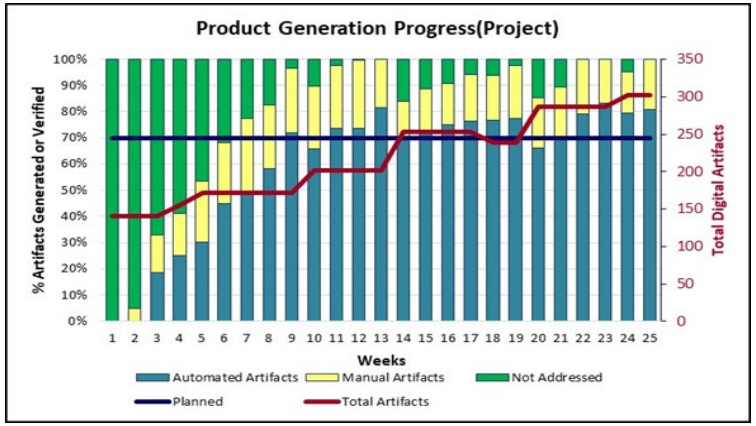
Adaptability and Rework





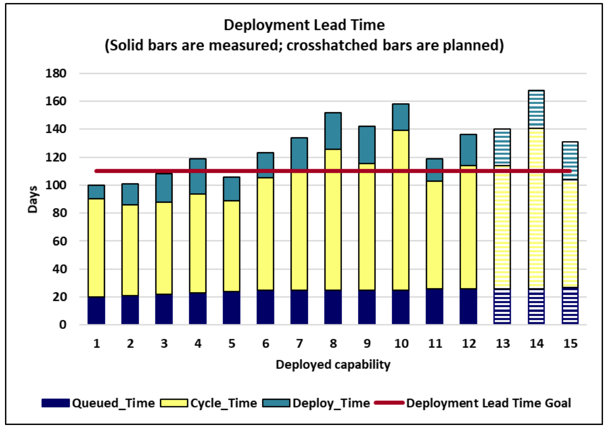
How much rework is for planned and unplanned changes?

Product Automation



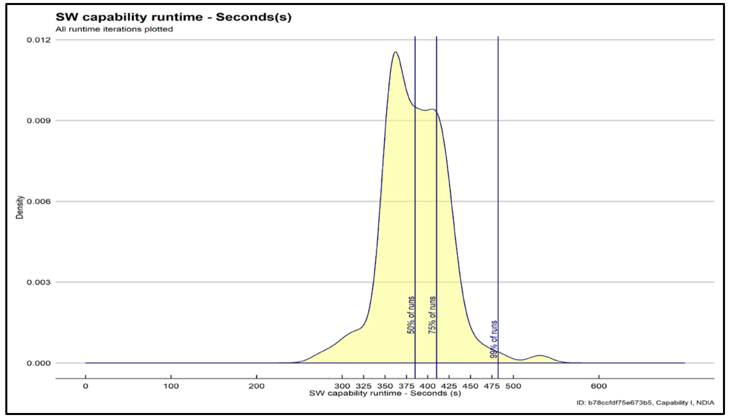
What percentage of artifacts are automatically model-generated?

Deployment Lead Time



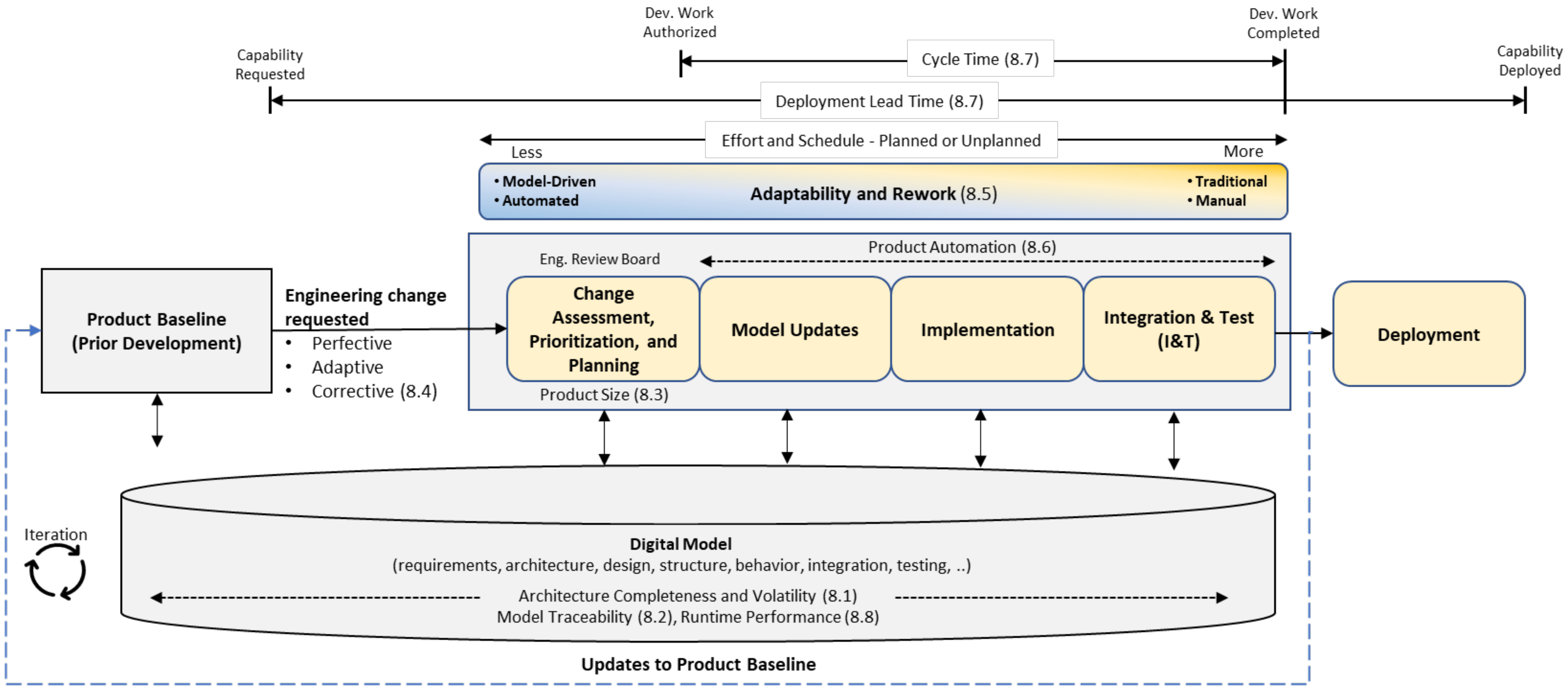
How long does it take to deploy an identified capability?

Runtime Performance



What is the likelihood performance will meet operational needs?

Tying it all together – DE measurement framework concept



Where do we go from here?

- *DE measures for the enterprise*
- *Measure return on investment*
- *Measure additional productivity indicators related to velocity and agility*
- *Measure additional indicators that isolate new value to the enterprise through DE, in areas such as quality and knowledge transfer*
- *Measure enterprise and personnel process adoption*
- *Measure breadth of usability and user experience with digital tools*
- *Supportability and maintainability measures (impact assessment agility)*
- *Measures for security*
- *Identify typical digital artifacts*
- *Specify leading indicators*

Much appreciation to the many individuals and organizations that supported development of the V1.0a Digital Engineering Measurement Framework!

Editors	Organization
Joseph M. Bradley	Main Sail, LLC Leading Change, LLC
Cheryl Jones	US Army DEVCOM AC Practical Software and Systems Measurement (PSM)
Geoff Draper	L3Harris Technologies AIA Engineering Management Committee NDIA Systems Engineering Division
Tom McDermott	Systems Engineering Research Center (SERC) Stevens Institute of Technology
Paul Janusz	US Army DEVCOM AC Practical Software and Systems Measurement (PSM)

Authors	Organization
Lennis Bearden	L3Harris Technologies
Giacomo Gentile	Collins Aerospace
Richard Halliger	Volkswagen Group
Ray Madachy	Naval Postgraduate School
Drew Miller	The Boeing Company
Young Park	Collins Aerospace
Bob Scheurer	The Boeing Company NDIA Systems Engineering Division AIA Engineering Management Committee

Core Team	Organization
Kaitlin Henderson	Virginia Tech
Al Hoheb	Aerospace Corporation
Bill Luk	BAE Systems AIA Engineering Management Committee
Lisa Murphy	Siemens Digital Industries Software
Jeff Nartatez	Office of the Undersecretary of Defense for Research and Engineering (OUSD R&E) [SAIC]
Garry Roedler	INCOSE NDIA Systems Engineering Division
Alejandro Salado	The University of Arizona
Frank Salvatore	Office of the Undersecretary of Defense for Research and Engineering (OUSD R&E) [SAIC]
Paul Segura	The Boeing Company AIA Engineering Management Committee
Natasha Shevchenko	Software Engineering Institute, Carnegie Mellon University
Marilee Wheaton	Aerospace Corporation INCOSE

Additional Contributors	Organization
David Allsop	The Boeing Company
Sanjay Angadi	Ansys AIA Engineering Management Committee
David Cooper	BAE Systems
Mimi Davidson	Office of the Undersecretary of Defense for Research and Engineering (OUSD R&E) [SAIC]
John Dilger	BAE Systems
Paul Embry	L3Harris Technologies AIA Engineering Management Committee
Ann Hodge	Sandia National Laboratories
Mike McLendon	Retired
Robert Minnichelli	Aerospace Corporation
Gery Mras	Aerospace Industries Association (AIA)
Rosco Newsome	Ernst & Young (EY)
Bill Nichols	Software Engineering Institute, Carnegie Mellon University
Ryan Noguchi	Aerospace Corporation
Macaulay Osaisai	L3Harris Technologies
Steven Quinn	BAE Systems
Donna Rhodes	Massachusetts Institute of Technology (MIT)
Chris Schreiber	Lockheed Martin Corporation NDIA Systems Engineering Division
Paul Solomon	Retired
Brian Tenney	Lockheed Martin Corporation AIA Quality Assurance Committee
Sundar Thyagarajan	L3Harris Technologies
Steven Turek	United States Air Force
Timothy Walden	Lockheed Martin Corporation AIA Engineering Management Committee
John Wood	Naval Information Warfare Command

Summary – Digital Engineering Measurement Framework v1.1

- Lack of common measures and established best practices have inhibited digital transformation
- The v1.1 release of the DE Measurement Framework establishes an initial consensus from our partners as a starting point to advance a discussion across industry – some measures are conceptual

This initial DE measurement framework proposed by our team of representative stakeholder experts is intended to help projects and enterprises establish an initial path toward a measurably effective transition and implementation of digital engineering methods. It is but the first steps along this path, it will be a long and challenging but rewarding journey, and our industry will learn, iterate, and evolve as we go. We hope enterprises across a variety of application domains will find this initial measurement guidance useful to assess the effectiveness of their respective digital engineering transformation initiatives.

- Help us improve it! Participate in reviews, provide comments and suggestions, pilot the measures proposed, and participate in the future evolution of this framework
- Contact our team leads to get further involved

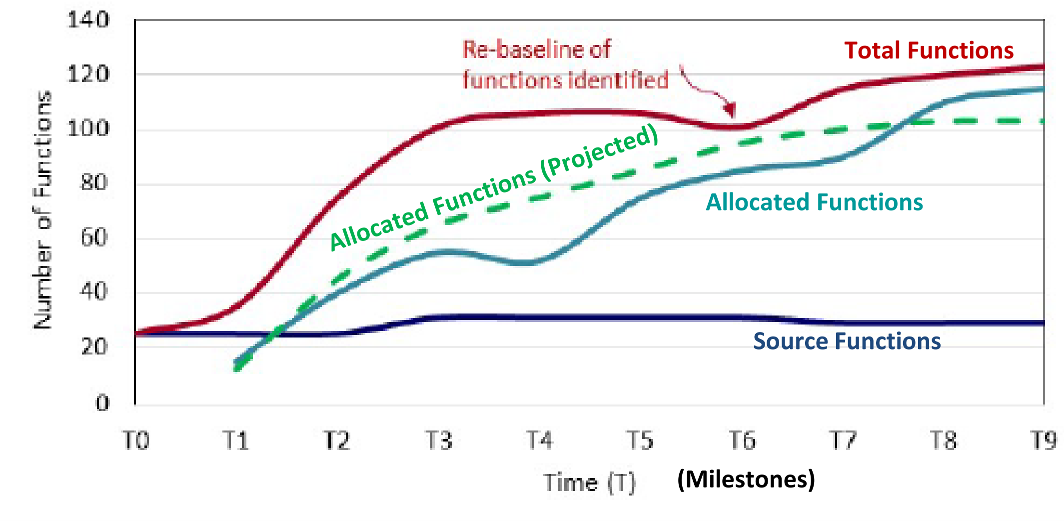
Joe Bradley
Leading Change LLC
josephbradley@leading-change.org

Cheryl Jones
Practical Software and Systems Measurement (PSM)
cheryl.l.jones128.civ@army.mil

Architecture Completeness and Volatility

Information Need	How complete is the architecture? Does the architecture account for all required functions? Is the architecture sufficiently complete to proceed with design at acceptable risk?

Example indicator – Functional Architecture Completeness



Functions Completed vs. Plan and Volatility Over Time
(Source Functions + Derived Functions)

Data Point	Source Functions	Derived Functions	Total Functions	Change Per Time	Allocated Functions	Functions Allocated (Remaining)	Function Volatility (Allocated) Per Time	Percent Functions Allocated
T0	25		25					
T1	25	10	35	10	15	20	15	0.43
T2	25	50	75	40	40	35	25	0.53
T3	31	70	101	26	55	46	15	0.54
T4	31	75	106	5	52	54	-3	0.49
T5	31	75	106	0	75	31	23	0.71
T6	31	70	101	-5	85	16	10	0.84
T7	29	80	109	8	90	19	5	0.83
T8	29	80	109	0	100	9	10	0.92
T9	29	80	109	0	106	3	6	0.97

Total = Source + Derived

Volatility = Changes / Time

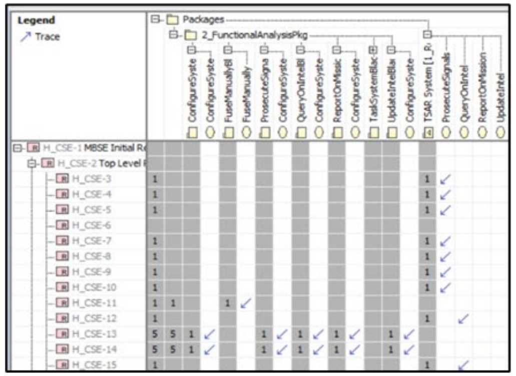
Total functions not yet allocated

Completeness = Functions Allocated / Total Functions

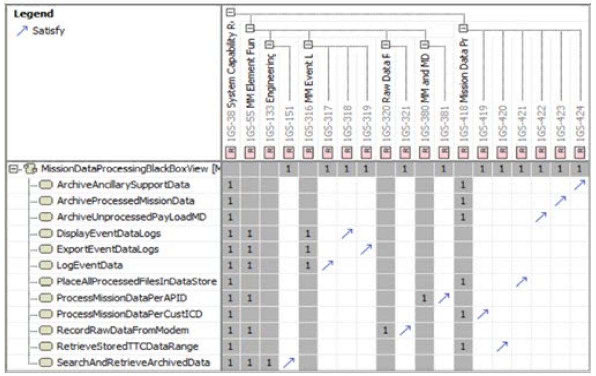
Evaluate iterative progress toward completion of an architecture (functional, logical, physical) based on allocated functions and interfaces. Completeness and stability of the architecture provides a direct view into the maturity of a system development.

Model Traceability

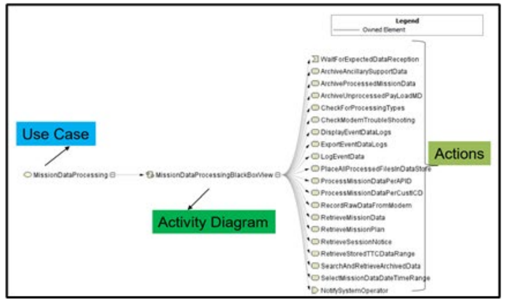
Information Need	What is the extent of achieved traceability coverage across model elements? What is the extent of traceability from requirements down to the logical or physical solution domain? What traceability gaps or defects exist in the digital model?



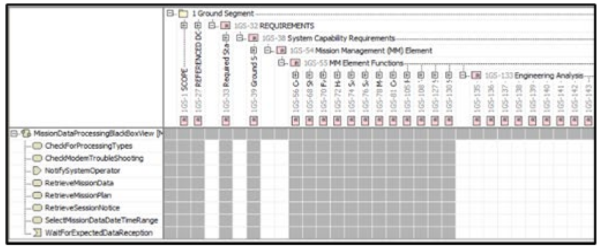
Traceability Between Model Elements (Dependency Matrix)



Relationships to Problem or Solution Domain («*satisfy*» or «*refine*» Matrix)



Relation Map Diagrams (Model Traceability, Ownership)



Identifying Model Traceability Gaps (Orphans)

Traceability reports and analyses are greatly facilitated by modern digital modeling tools. The traceability concepts and indicators in this specification are representative examples of more general traceability mappings and reports across the development life cycle, such as:

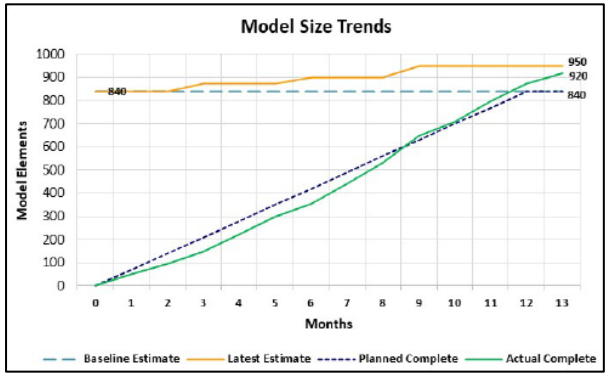
- Traceability between stakeholder needs, system requirements, and allocated or derived requirements at each level of the system hierarchy
- Traceability and flow down of requirements to the logical or physical solution domain (e.g., design, implementation, integration, verification, validation)
- Allocation and traceability of performance measures or parameters, such as Measures of Effectiveness (MOEs) or Key Performance Parameters (KPPs)
- Traceability of system interfaces

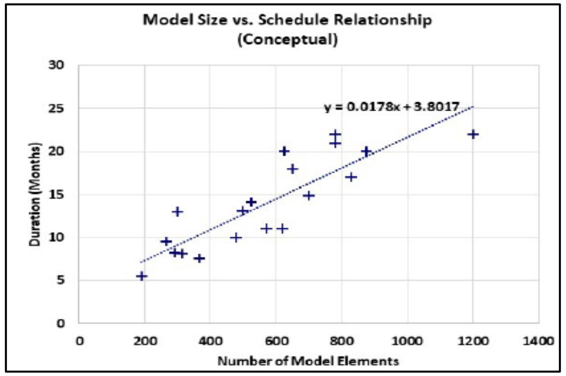
<u>Excerpts from</u>: 'MBSE and Requirements Analysis, key to Successful System Engineering', M. Osaisai and F. Markham, 2019 MBSE Cyber Experience Symposium.

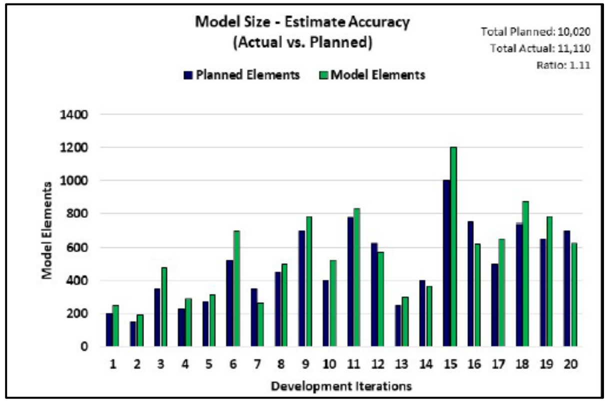
Assess traceability between modeling elements to assure allocation, flow down, and coverage.
Evaluate gaps to assure the system solution is complete and consistent.

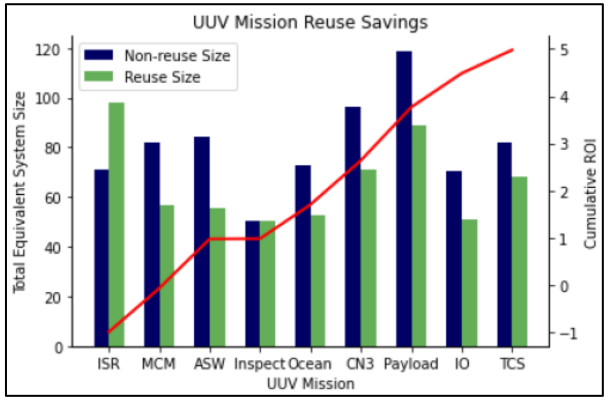
Product Size

Information Need	What is the size and scope for the digital engineering project or product? How much work must be done? How does product size relate to estimates and measures of cost, schedule, productivity, performance, or ROI?









Product Size is initially proposed as a count of model elements (pilot).

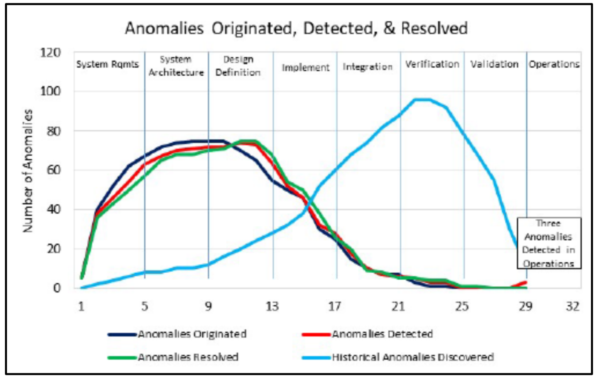
Several candidate measures could be derived from product size measures, such as these below.

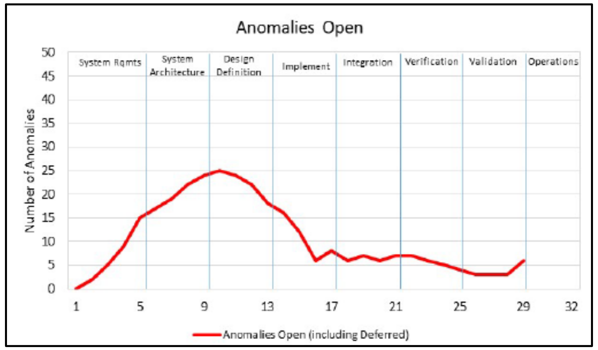
- Productivity = (Product Size) / (Effort)
  Number of model elements generated per unit effort (e.g., model elements / labor hours)
- Progress = (Product Size $_{\text{actual to date}}$) / (Product Size $_{\text{planned}}$)
  Proportion or percentage of planned model elements completed for characterizing progress and work remaining. Can also be used to characterize growth and stability (actual size vs. plan).
- Throughput = (Product Size) / (Duration)
  Number of model elements completed per calendar period, e.g., elements / month. Can also be used to characterize a size vs. schedule relationship.

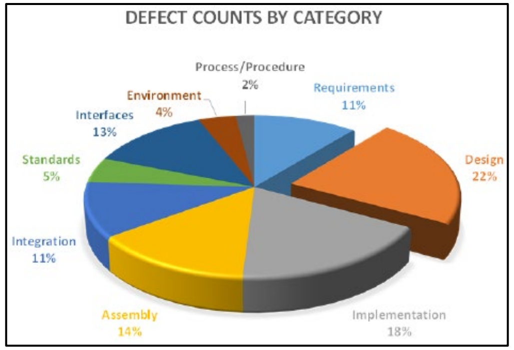
Product size can be used as a proxy for deriving other measures (e.g., effort, schedule, productivity, capability).
Currently proposed as a count of model elements to help advance further industry discussion.

DE Anomalies

Information Need	Is the quality of the product adequate to be used in subsequent phases or activities? Are we finding and removing anomalies early in the life cycle using models and shared information? Is the use of DE leading to detection of anomalies earlier in the lifecycle compared to traditional methods or projects? How can DE and modeling efforts be improved to reduce the leading causes of anomalies?







Improved system quality and early defect detection are primary benefits expected from DE.

- *Anomaly* is the term used to discuss deviations from expectations (e.g., errors, warnings, defects, change orders, problem reports, corrective actions)

Anomalies are collected, analyzed, and monitored across lifecycle activities or boundaries (e.g., stages, phases, iterations, releases)

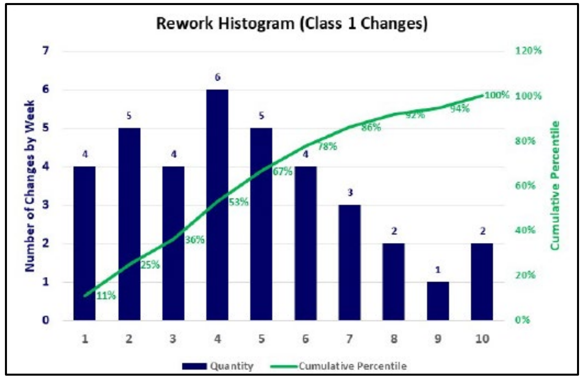
Analyze attributes of anomalies to enable root cause analysis and improvement actions, such as:

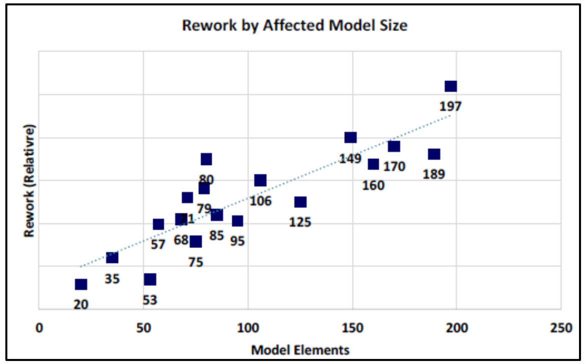
Date timestamps (opened, closed, state transitions)	Anomaly state (open, in work, resolved, closed, …)	Work activity (originated, detected, resolved)
Work product type	Product identifier	Anomaly category
Severity	Customer impact	Rework effort

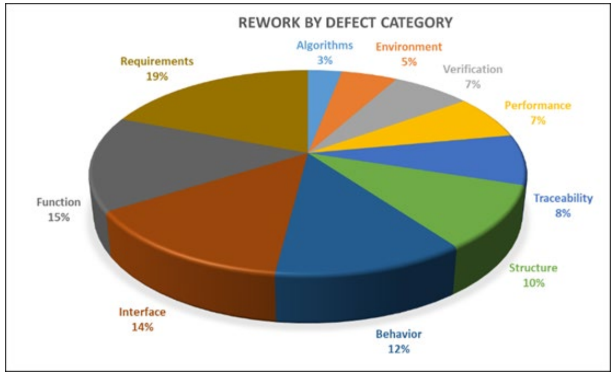
Measure the effectiveness in timely detection and removal of anomalies during development (saves) vs. down stream activities (escapes).

Adaptability and Rework

Information Need	How much rework effort is spent maintaining planned or unplanned changes to digital engineering work products across the life cycle? Can changes to engineering work products be implemented more easily and with less effort in a digital engineering environment relative to traditional methods?







Traditionally, rework measures are focused on the effort to implement corrective actions for repair of defects. Here we envision the broader use of rework measures enabled through digital engineering to include change management, adaptability, and impact assessment contexts beyond simply the correction of defects.

- Corrective actions – repair of anomalies or defects
- Perfective actions – planned or scheduled enhancements
- Adaptive actions – adapting configurations to other environments

Typically driven by change requests, under the governance of a Configuration Control Board (CCB) or equivalent.

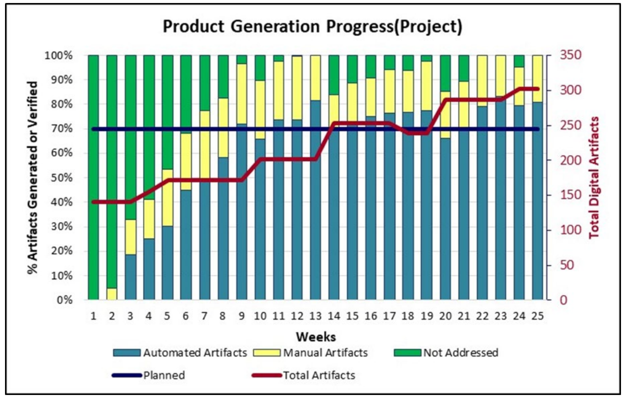
Model-driven products can be more resilient to changes with reduced rework impact

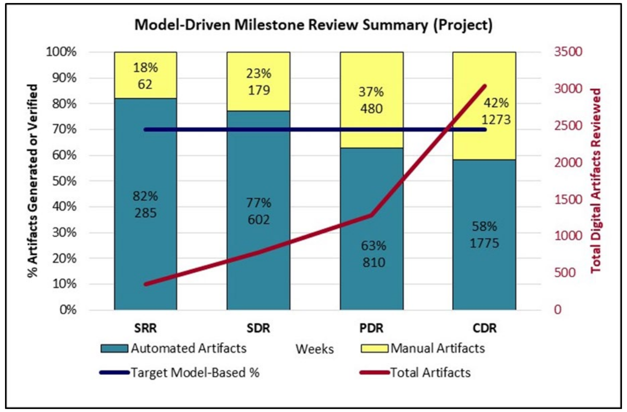
- Analyze rework distributions (effort; cost; schedule; resources) for a set of changes or attribute types

Measure how readily changes (planned and unplanned) can be implemented.
Model-based products can be more resilient to changes and facilitate automated product updates.

Product Automation

Information Need	What percentage of artifacts are automatically generated from digital models? To what extent are artifacts facilitating program reviews? How much is automation contributing to meeting performance and quality objectives?





Model-driven development provides opportunities to automate engineering processes and generation of work products that have often been done manually in traditional approaches.

Model-based work products such as requirements, architecture, design, use cases and other views or modeling artifacts can be automatically generated and published directly from modeling tools, at significant savings in effort relative to traditional documentation-centric approaches.

Examples:

- % of digital model artifacts produced via automation
- % of requirements verified through automation of digital model parameters and constraints
- % of labor hours spent generating digital artifacts through automated vs. manual methods

Potential benefits:

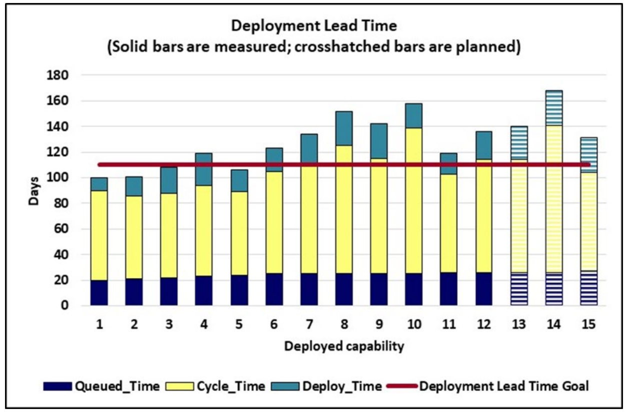
- Process efficiencies. Labor reductions. Shorter cycle times. Less rework. Earlier V&V of solutions.
- Automated model-based generation of milestone review artifacts.

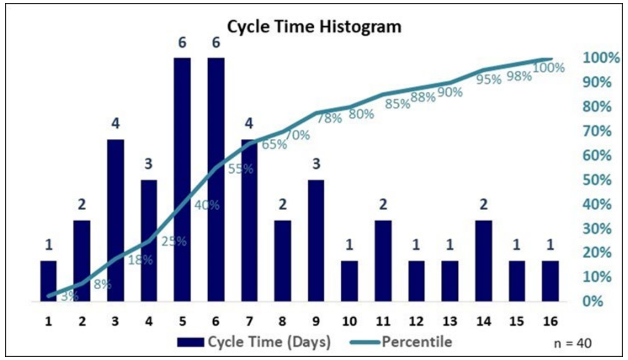
Objectives for the extent of model-driven automated artifact generation may be specific to the product or domain. Automation in the range of 70%-80% is often beneficial in producing improved performance outcomes, but this may vary by domain or application.

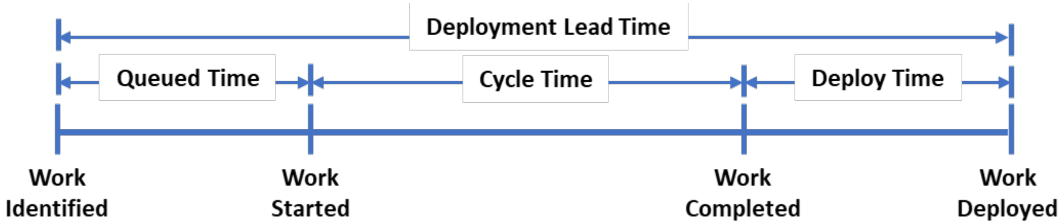
Measure the extent to which work products and reviews can be automated through digital models

Deployment Lead Time

Information Need	How long does it take to deploy an identified feature or capability? How long does it take to deploy a viable product for operational use after a request is received? Where is the deployment bottleneck; in planning/backlog, implementation, or deployment of the implemented capability?







Shorter deployment lead times and cycle times can indicate more efficient delivery/deployment flow and quicker response to business objectives or mission needs. Longer deployment lead times and cycle times are often correlated to the scope, product size, and complexity of work products.

Attributes characterizing the relative work performed (e.g., product requirements, model elements, product size, complexity) can be used to normalize and synthesize comparable work performed under similar defined conditions.

Oftentimes, deployment requires coordination with the acquirer or operational environment outside the supplier’s control. From the supplier’s perspective, potential delays in scheduling access to the operational environment can greatly affect overall Deployment Lead Time. For these reasons, measures based on Deploy Time can be interesting and useful to some extent but may be not as repeatable or actionable as Cycle Time which is more under direct project control.

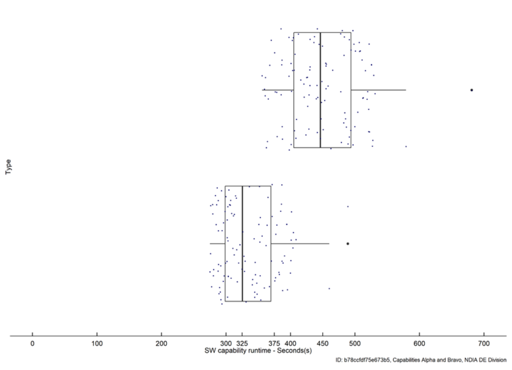
Under consistent conditions, deployment lead time can be used as a measure of team capability and throughput.

Measure how rapidly authorized system capabilities can be engineered, developed, and delivered for use in their intended operational environment.

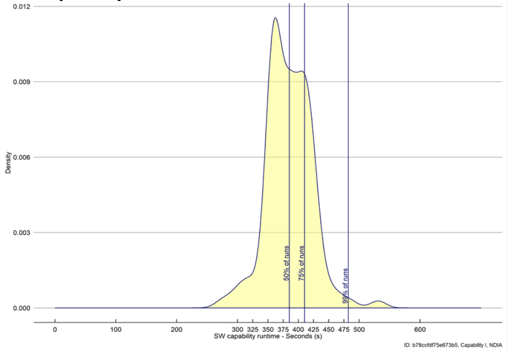
Runtime Performance

Information Need	What is the runtime performance of the capability or system? What is the likelihood that runtime performance will meet operational requirements? Where are the runtime performance bottlenecks, and how can operational performance be optimized?

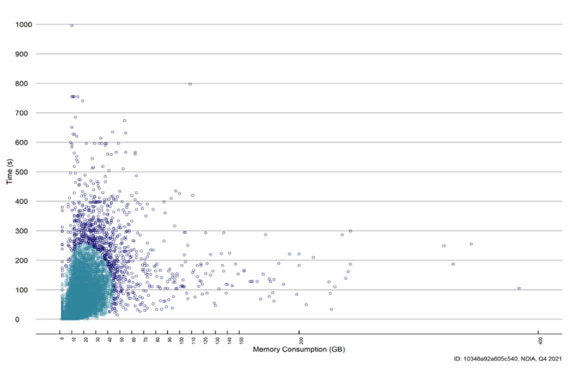
SW Capability Runtime - duration in second(s)



SW Capability Runtime - distribution



Performance Benchmark



Performance analysis is critical to early requirements development, architecture, and design processes to ensure the ultimate target solution is feasible. This is generally done through sophisticated models, simulations, and prototypes to validate applicable algorithms or ranges of performance prior to final implementation and deployment in the operational environment.

The tech stack hosts models that form a digital twin.

Runtime performance is a particular concern for models that tax the computing infrastructure, where data latency or sluggish infrastructure performance can have significant adverse effects on the digital design effort.

Performance analyses can be plotted to tailor future capabilities to their expected environments and workloads.

Measure the time it takes a software system to perform or execute capabilities or assess alternative model-based solutions.