Measuring MBSE Model Maturity
A Library-Supported Approach

Presented by Matthew Wylie, Shoal Group, on behalf of Brett Morris and Derek Rogers

SYSTEMS ENGINEERING TEST & EVALUATION
SETE 2024

22 - 25 SEPTEMBER 2024

Melbourne, Australia

ENGINEERS AUSTRALIA | Systems Engineering Society of Australia

INCOSE

ITEA

Simulation Australasia

Introduction


Background


Existing approaches and key findings


Proposed MBSE Model Maturity Assessment Method


Building a Library of MBSE Maturity Measures


Conclusions

Background

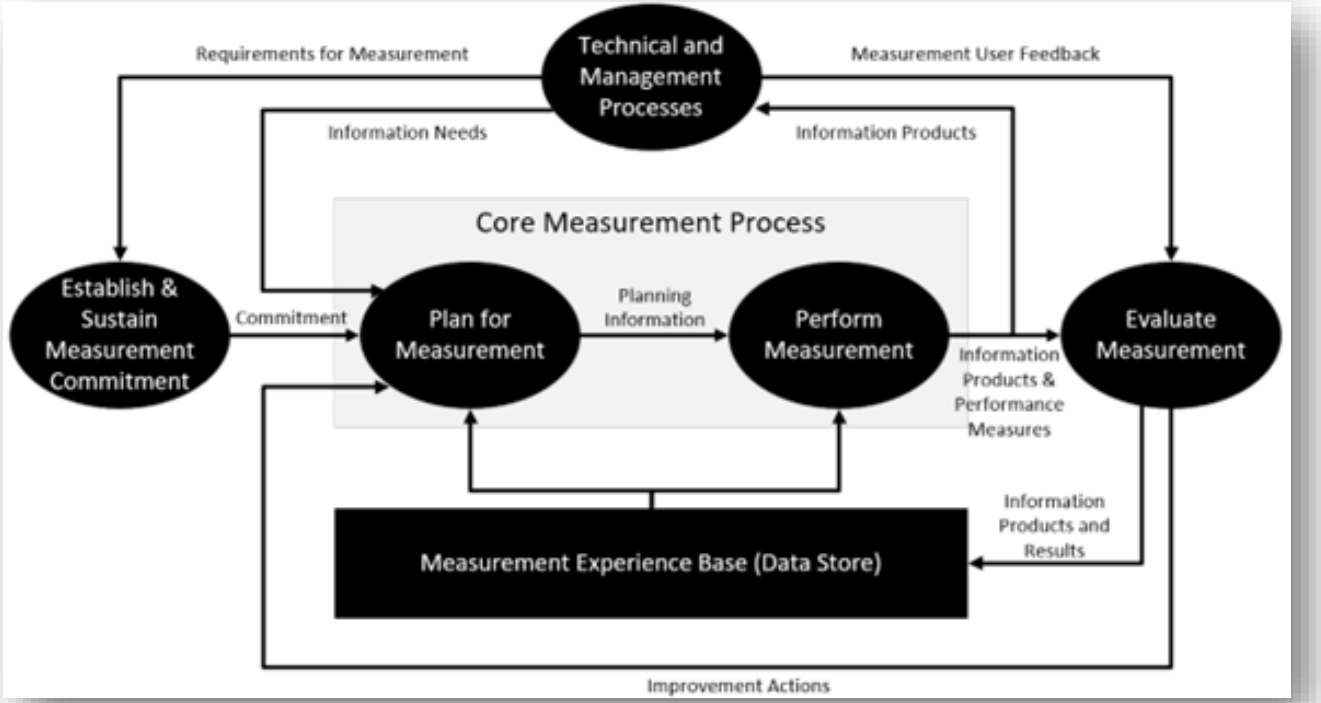
Background

- Modelling without stopping criteria is problematic
- Efficient and effective modelling requires an understanding of modelling purpose and goals
- Tracking and assessing maturity
  - Essential for managing resources and ensuring clients' needs are met
  - Often reliant on expert judgement, assisted by model validation tools

Background

- How can the maturity of an MBSE model be measured?
- What attributes of an MBSE model provide an indication of model maturity?

- Starting point:
  - ISO/IEC/IEEE 15939
  - Measurement Process Model



ISO/IEC/IEEE 15939 Systems and Software Engineering - Measurement Process, 2017

Existing approaches and key findings

SHOAL™

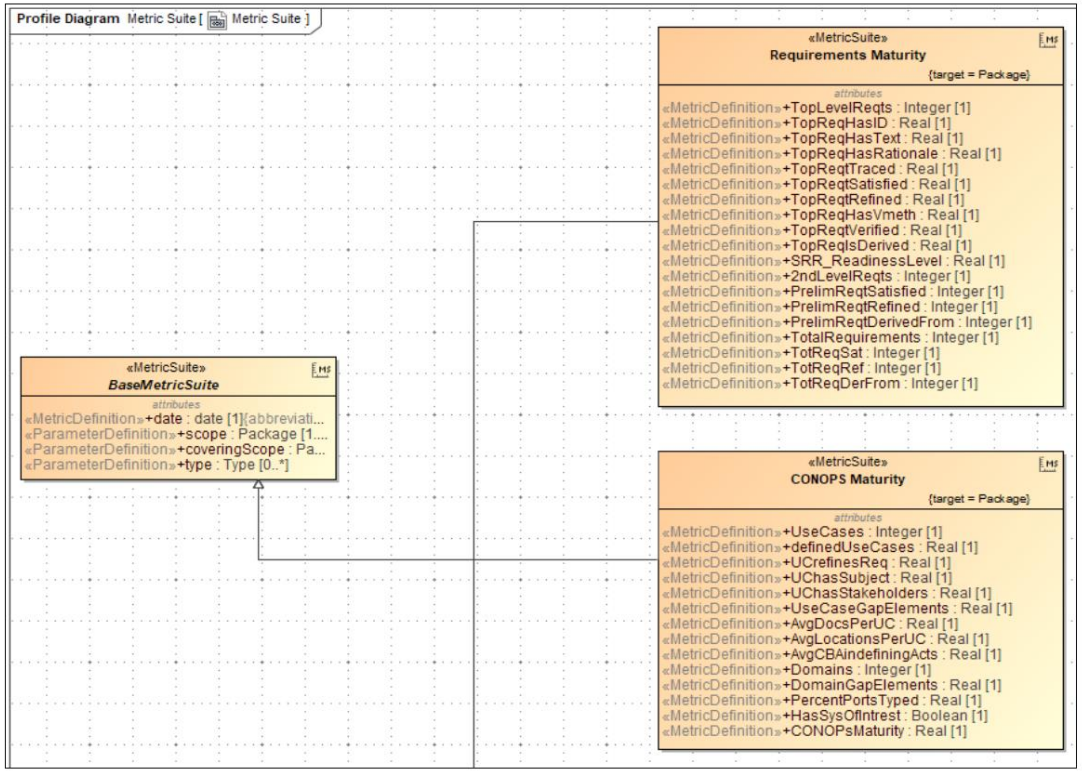
Measuring Model Maturity with Checklist Criteria

- Approach focuses on maturity metrics for Mandated System Review (MSR) entry criteria (or other goals).
- Measure % complete in three areas of the model:
  1. Relationship Mapping
  2. Metamodel
  3. Methodology
- Three steps to developing MBSE model maturity metrics:
  1. Define modelling maturity attributes.
  2. Construct metric suites using the maturity attributes.
  3. Capture and assess metrics.

Measuring Model Maturity with Checklist Criteria

- Leverage Cameo's <<MetricSuite>>
- Importance of metamodel.

#	Name	Date	Detail Design Comp Lvl	Top Level Reqts	Top Reqt Satisfied	Top Reqt Traced
1	Current Requirements	2018.11.01 17.53	100	16	16	16
2	2018 10 31	2018.11.01 17.53	100	16	16	16
3	2018 10 10	2018.10.10 09.42	94.6237	16	16	16
4	2018 10 08	2018.10.08 10.17	94.6237	16	16	16



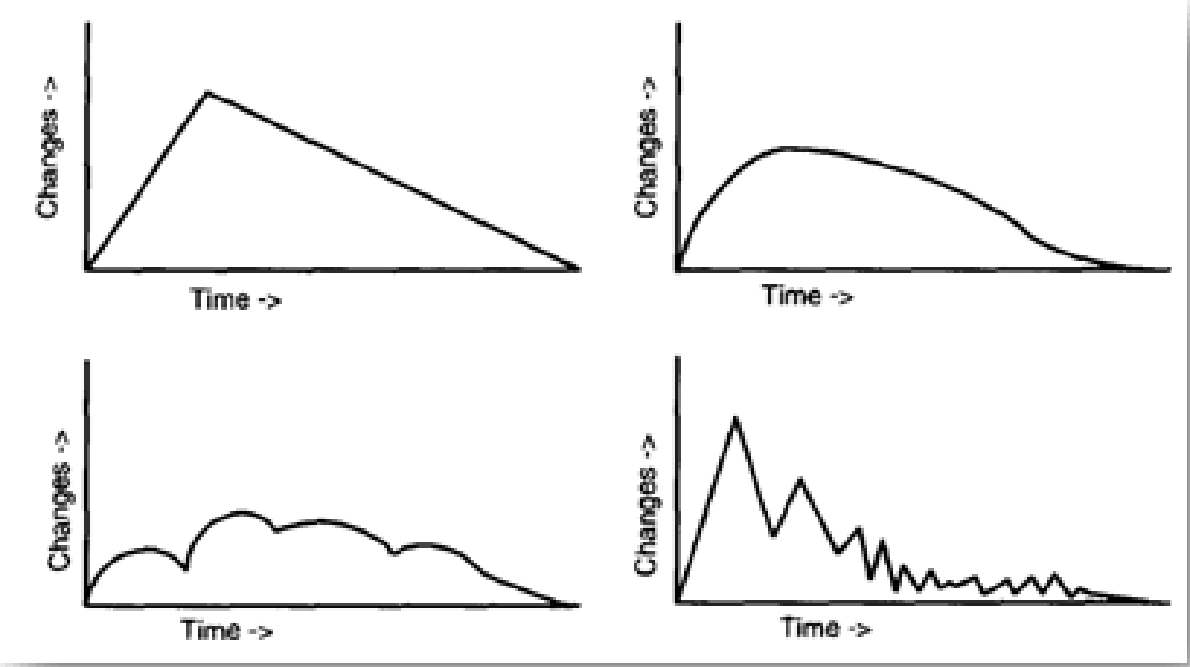
Gaskell, J. D., & Harrison, C. N. (2019). Improved System Engineering Technical Review's Entrance/Exit Criteria with Model Maturity Metrics. ISSE 2019 – 5th IEEE International Symposium on Systems Engineering.

Measuring Model Maturity with Checklist Criteria

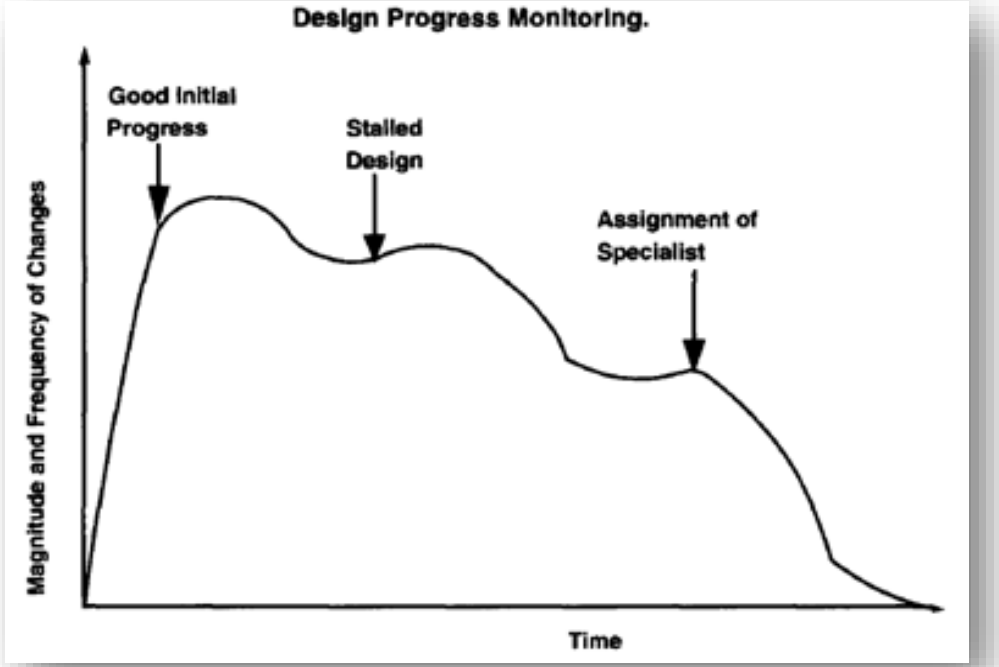
- Key takeaways from this approach:
  1. A well-defined metamodel is essential.
  2. Maturity metrics don't assess gaps, or quality of the modelled elements.
  3. Very useful approach for programs with linear lifecycle models with defined milestones.
     - Not all Shoal modelling projects use such a lifecycle
     - A more flexible approach is required

Measuring Model Maturity by Volatility

- Approach is based on premise that the number and frequency of design changes reduces as the design matures:



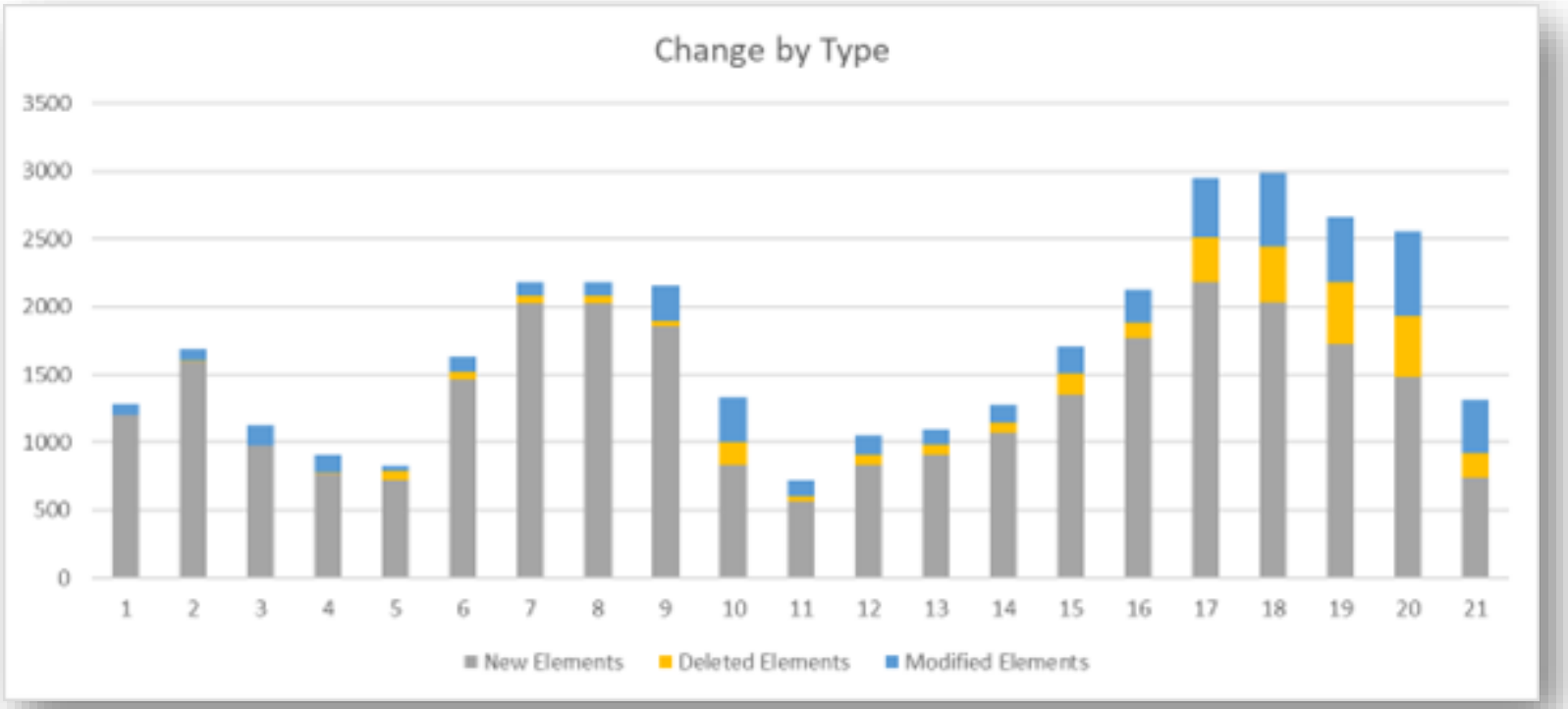
Design changes vs. time as an indicator of design maturity (O'Brien & Smith, 1995)



Design change profile vs. time with possible sources of peaks and troughs (O'Brien & Smith, 1995)

Measuring Model Maturity by Volatility

- MBSE Model changes can be extracted easily from a model.



- However, there are many possible sources for change, e.g. changes in work priorities, staff availability, IT issues, changing user requirements, etc.

Measuring Model Quality as a Surrogate for Maturity

- Approach based on V&V of models.
- Easier with simulation models – can quantify the results.
- Not so easy with MBSE models as they are descriptive.
- Model Quality Metrics:
- How do we measure these attributes?

Paige	Friedenthal	ISO/IEC 9126
Simplicity	Purpose well-defined?	Functionality
Uniqueness	sufficient scope	Reliability
Consistency	model fidelity	Usability
Seamlessness	completeness relative to scope	Efficiency
Reversibility	well-formed	Maintainability
Scalability	internal consistency of the model	Portability
Supportability	understandability of the model	
Reliability	accurate or valid model of domain of interest	
Space economy		

Descriptive Model Quality Criteria from Various Sources (Giachetti, 2017)

Model-Based Architecture Assessment

- Approach focuses on the model connectedness:
- Maturity assessed through combination of axioms:

Phase 1 Model Maturity		
1.1	No action generates and receives the same input/output.	$(\forall a \in A)(\neg\exists n \in N)[generates(a,n) \wedge receives(a,n)]$
1.2	If any action generates an input/output, it also receives an input/output.	$(\forall a \in A)(\exists n_1 \in N)[generates(a,n_1) \rightarrow (\exists n_2 \in N) receives(a,n_2)]$
Phase 2 Model Maturity (Phase 1 axioms, plus the following are satisfied)		
2.1	Any input/output is generated by some action.	$(\forall n \in N)(\exists a \in A)[generatedby(n,a)]$
2.2	Any input/output is received by some action.	$(\forall n \in N)(\exists a \in A)[receivedby(n,a)]$
2.3	Any action is performed by some asset.	$(\forall a \in A)(\exists p \in P)[performedby(a,p)]$
2.4	Each action generates or receives at least one input/output.	$(\forall a \in A)(\exists n \in N)[generates(a,n) \vee receives(a,n)]$
2.5	Each asset performs at least one action.	$(\forall p \in P)(\exists a \in A)[performs(p,a)]$
Phase 3 Model Maturity (Phase 2 axioms, plus the following are satisfied)		
3.1	Each asset is connected by at least one conduit.	$(\forall p \in P)(\exists c \in C) connectedby(p,c)$
3.2	Any conduit connects to at least two disjoint assets.	$(\forall c \in C)(\exists p_1 \in P)(\exists p_2 \in P)[connectsto(c,p_1) \wedge connectsto(c,p_2) \wedge (p_1 \neq p_2)]$
3.3	Any conduit connects to no more than two assets.	$(\forall c \in C)\ (\neg\exists p_1 \in P)(\neg\exists p_2 \in P)(\neg\exists p_3 \in P)$ $[connectsto(c,p_1) \wedge connectsto(c,p_2) \wedge connectsto(c,p_3) \wedge (p_1 \neq p_2) \wedge (p_2 \neq p_3) \wedge (p_1 \neq p_3)]$
Phase 4 Model Maturity (Phase 3 axioms, plus the following are satisfied)		
4.1	If any two assets exchange some input/output, those assets are connected to at least one common conduit.	$(\forall p_1 \in P)(\forall p_2 \in P)(\exists n \in N)$ $[exchanges(p_1,p_2,n) \rightarrow (\ (\exists c \in C) connectsto(c,p_1) \wedge connectsto(c,p_2)\ )]$
4.2	Every exchanged input/output between any two assets is transferred by some conduit that connects to those assets.	$(\forall p_1 \in P)(\forall p_2 \in P)(\forall n \in N)$ $[exchanges(p_1,p_2,n) \rightarrow$ $(\ (\exists c \in C) transferredby(n,c) \wedge connectsto(c,p_1) \wedge connectsto(c,p_2)\ )]$
Phase 5 Model Maturity (Phase 4 axioms, plus the following are satisfied)		
5.1	Each asset generates an input/output to or receives an input/output from at least one other disjoint asset.	$(\forall p_1 \in P)(\exists p_2 \in P)(\exists n \in N)[\ (produces(p_1,n) \wedge consumes(p_2,n)\ ) \vee$ $(\ consumes(p_1,n) \wedge produces(p_2,n)\ ) \wedge p_1 \neq p_2\ ]$

Example Model Maturity Axioms from (Giammarco K. , 2014)

A Digital Engineering Measurement Framework

- The INCOSE DE Measurement Framework provides a useful set of measures and measurement process for DE maturity.
  - Many of these could be applicable to MBSE model maturity:

Architectural Completeness	Model Coverage	Functional Size and Stability
Functional Correctness	Process Effectiveness	Process Efficiency
Automation	Speed	Technology Performance

- Framework provides a set of potential measures and their descriptions, information needs, indicators, etc.

A DE Measurement Framework (cont.)

Digital Engineering Measurement Framework

- Focused on measuring DE maturity,
- But … provides an information model and some useful measures for MBSE model maturity.
- Like Gaskell & Harrison – highlight need for defined metamodel to underpin measurement.

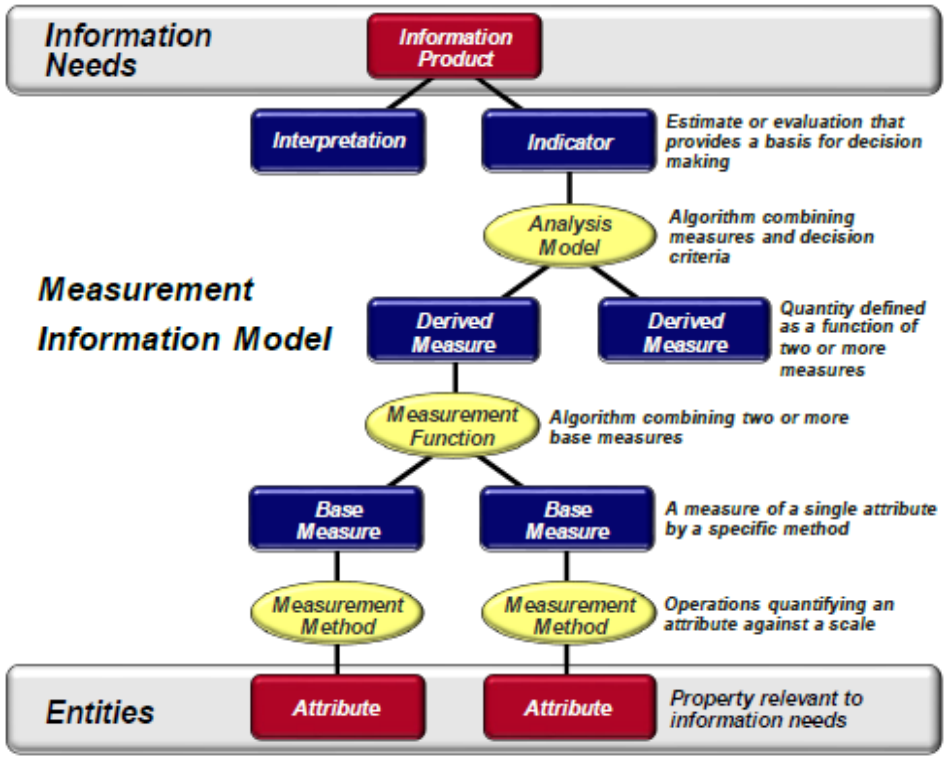


Figure 4-3: Measurement Information Model

Information Category	Measurable Concepts	Potential Measures
Schedule and Progress	Architectural Completeness	Architecture Completeness and Volatility
Schedule and Progress	Model Coverage	Model Traceability Model Coverage (e.g., modelled elements
Size and Stability	Functional Size and Stability	Product Size (e.g., Model Elements) Architecture Completeness and Volatility Functions Identified Functional Change Requests
Product Quality	Functional Correctness	DE (i.e., MBSE) Anomalies
Product Quality	Functional Correctness	Adaptability and rework Acceptance of Completed Work (e.g., Model Elements, Artifacts Rework or rework defects
Product Quality	Functional Correctness	Model Traceability Traceability Anomalies
Process Performance	Process Effectiveness	Model Element DE (i.e., MBSE) Anomalies
Process Performance	Process Effectiveness	DE (i.e. MBSE) Anomalies Rework Effort Reworked Model Elements
Process Performance	Process Efficiency - Automation	Product Automation Cycle Time
Process Performance	Process Efficiency - Speed	Deployment Lead Time Cycle Time
Process Performance	Process Efficiency	Productivity Model Elements/Release Artifacts/Release
Technology Effectiveness	Technology Performance	Runtime Performance Elapsed Time

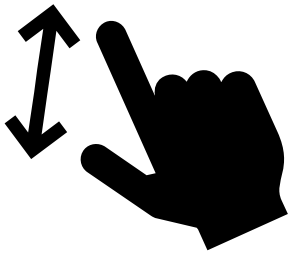
INCOSE. (2022). Practical Software and Systems Measurement (PSM) Digital Engineering Measurement Framework. INCOSE-TP-2022-002-01

Key Findings

- The approach of working backwards from the required model outputs is useful.
- A defined and well implemented metamodel with explicit model element classes is a necessary enabler of model maturity measurement.
- Model change volatility appears to provide an indication of model maturity --> potentially unreliable.
- Quality metrics difficult to measure in a quantitative manner.
- A library-based approach (e.g. INCOSE DE Measurement framework) would be a useful starting point.

Key Findings – Gap in MBSE Model Maturity Assessment

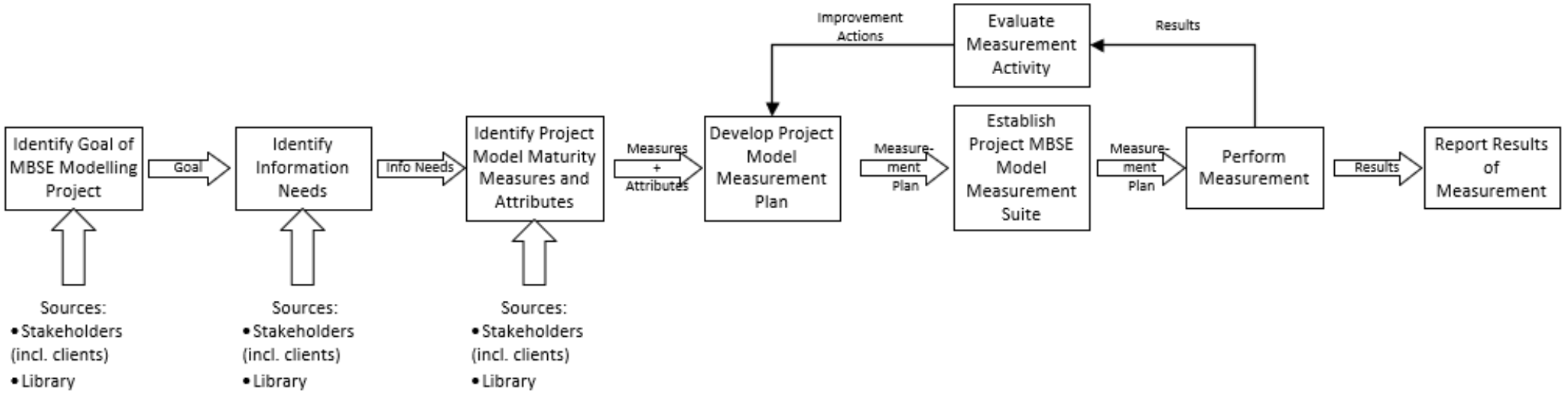
- The gap that needs to be addressed for measuring MBSE model maturity on projects is to build a flexible method that can adapt to the different program lifecycles and types of architectures (i.e. operational, functional, physical, etc.) being modelled.
- Ideally, model maturity metrics would be calculated from within the model.



Proposed MBSE Model Maturity Assessment Method

SHOAL™

Proposed MBSE Model Maturity Assessment Method



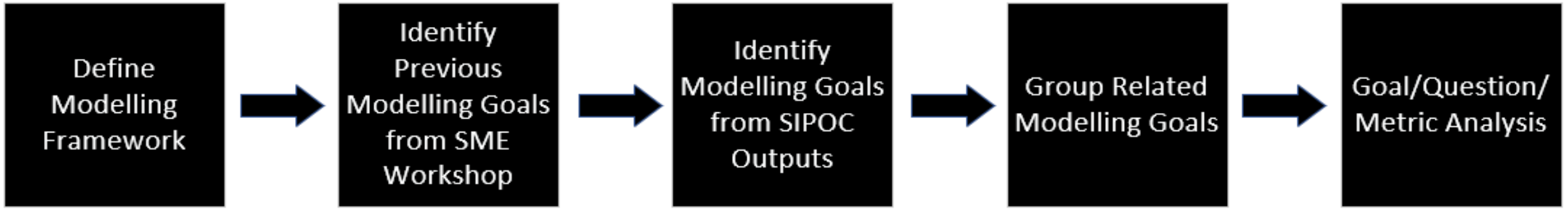
Extension of the INCOSE DE Measurement Framework measurement process

Building a Library of MBSE Maturity Measures

SHOAL™

Process overview

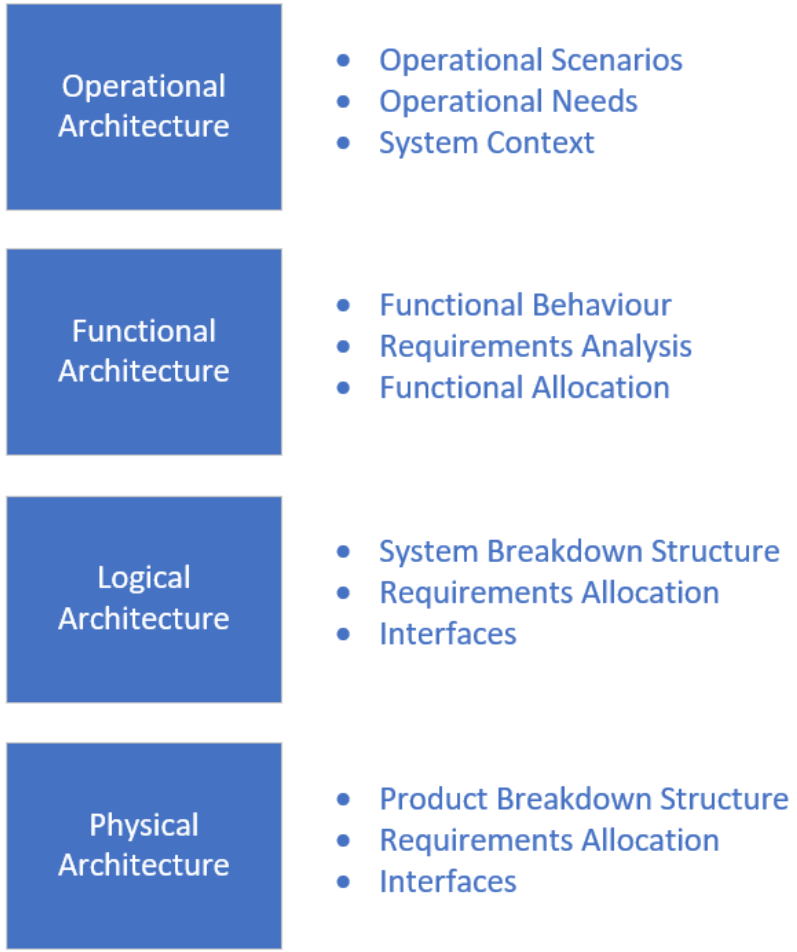
- Needed a process to generate MBSE model maturity measures
- Start by defining a modelling framework to facilitate the development of a structured, reusable set of metrics…



MBSE Maturity Model Library Development Process

Modelling Framework

- Developed through Shoal SME discussions
- Represents typical architecture levels and activities of Shoal modelling projects.
- Consistent with UAF®, DoDAF, and the NATO Architecture Framework (NAF)



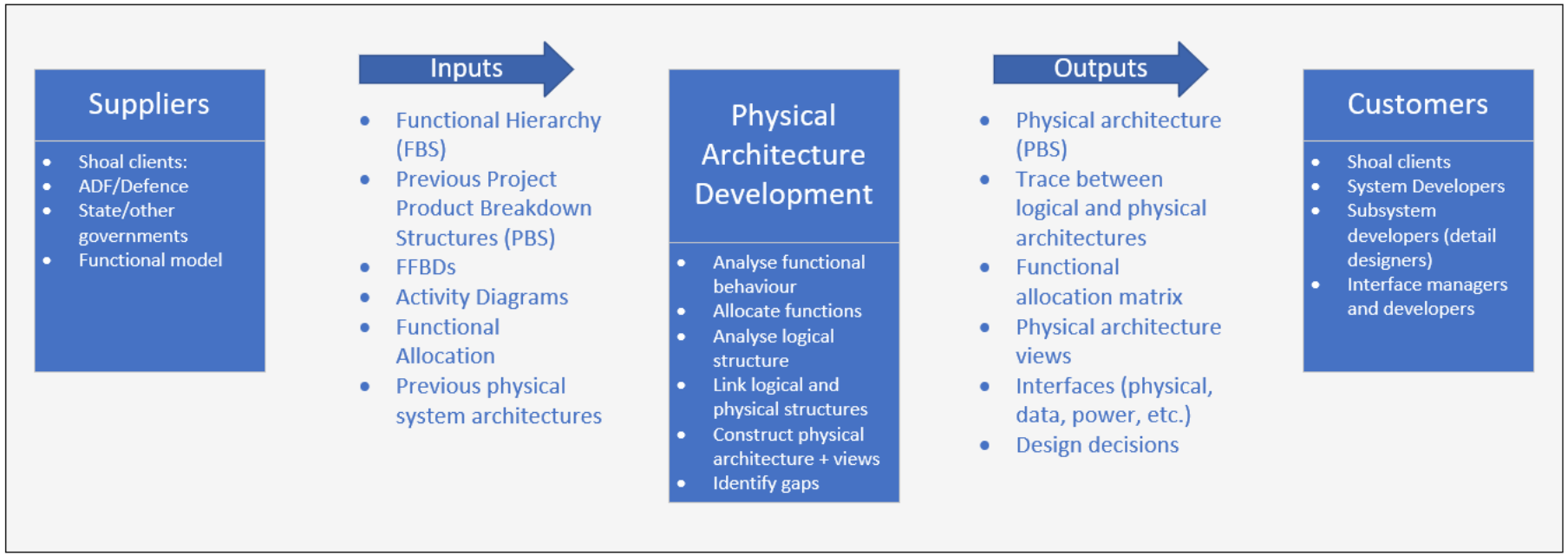
Architecture Levels for Grouping Modelling Goals and example modelling activities

Previous Modelling Goals

- Some previous goals of Shoal MBSE modelling projects included:
  - Operational level – to generate a complete set of Measures of Effectiveness (MOEs).
  - Functional level – to generate a complete set of Measures of Performance (MOPs).
  - Logical level – to develop a System Breakdown Structure (SBS).
  - Physical level – to develop a Product Breakdown Structure (PBS).

SIPOC Diagrams

- Suppliers, Inputs, Processes, Outputs, Customers.
- Used to generate additional modelling goals for each architecture level.
- Example for the Physical Architecture level:



Physical architecture level MBSE modelling SIPOC diagram

Grouping Modelling Goals

- Grouped previous modelling goals with those identified using SIPOC analysis.
- Provided validation of SME's goals.
- Identified some extras.
- Example for the Physical Architecture level.

Modelling Level	Modelling Goals (Workshop)	Modelling Goals (SIPOC Outputs)
Physical	Develop Breakdown Structures	Product Breakdown Structure (PBS)
	Assist in Requirements Definition	Requirements allocated to systems
		Requirements gap analysis
	Other	Develop physical architecture views
		Identify physical interfaces

Physical Architecture Level Modelling Goals from Workshop and SIPOC Diagrams

Goal Question Metric Analysis

- GQM approach roots in software development.
- Used grouped modelling goals as the input to GQM analysis.
- Analysed all four levels of the modelling framework.
- Library of maturity measures generated.
- Example for the Physical Architecture level.

Modelling Level	Modelling Goals (SIPOC Outputs)	Question?	Model Maturity Metrics
Physical	Product Breakdown Structure (PBS)	How many levels are complete in the PBS?	Levels completed
		How complete is the system's PBS?	% lowest level PBS elements allocated to Configuration Items
	Requirements allocated to systems	How complete is the allocation of requirements to the phys. arch?	% unallocated requirements
	Requirements gap analysis	How many requirements/system elements are unallocated?	% Systems without trace to requirements
	Develop physical architecture views	How many of the required physical arch. views are complete?	% required physical arch. views complete
	Identify physical interfaces	Have all physical interfaces been identified?	% physical interfaces identified
	Trace between logical and physical arch.	How complete is the trace from logical to physical architectures?	% physical elements traced from logical elements

Outputs from a GQM analysis for the physical architecture level modelling goals

Conclusions

SHOAL™

Summary

- Several efforts have been undertaken to create methods to measure model maturity in different domains
  - none of these were directly suitable for Shoal's MBSE modelling projects.
- We have developed a process to generate a library of MBSE model maturity measures for Shoal's projects.
  - Leveraged SMEs, SIPOC and GQM.
- Approach could be used by any organisation undertaking modelling projects.

SHOAL™

Questions and comments

MATTHEW WYLIE

SHOAL™

107 WRIGHT STREET, ADELAIDE SA 5000

AUCKLAND | BRISBANE | CANBERRA | MELBOURNE | SYDNEY

+61 2 6239 4288

support@shoalgroup.com

shoalgroup.com

Shoal Group Pty Ltd