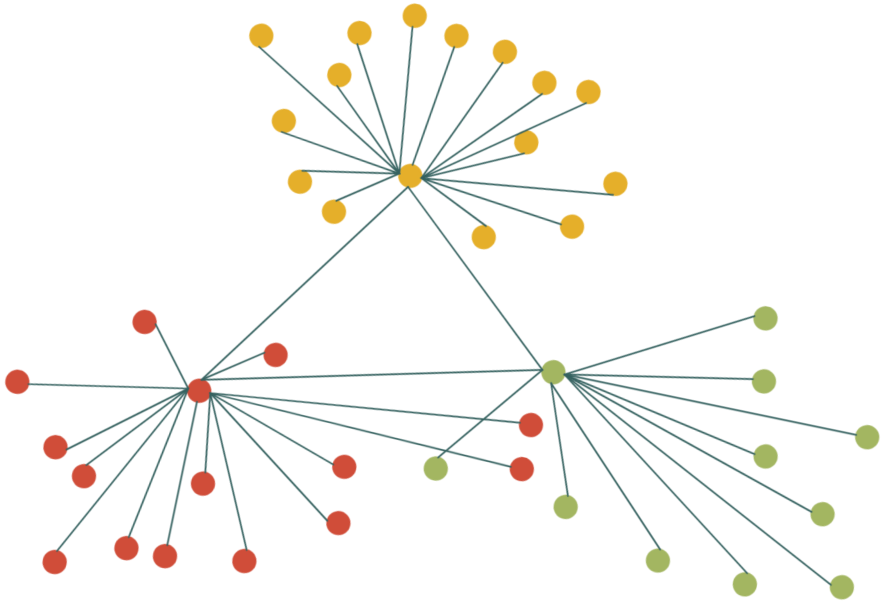
Managing Engineered Consistencies: Reconciling Semantics of Confirmation Frameworks



Encouraging A Conversation Across Technical Societies

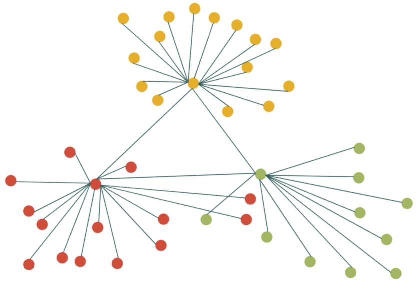
schindel@ictt.com

Discussion Draft
V1.2.4

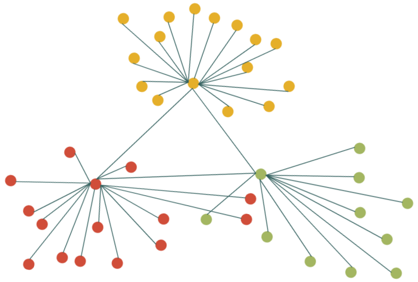
Purpose and scope

- The following material summarizes a growing challenge to disciplined engineering outcomes, and a recommended strategy for addressing it.
- This perspective is based upon a number of years of effort across several disciplines and the work of several technical societies.
- The current draft is limited to a summary level argument and strategy, for consideration by groups in several societies weighing a more active recommended collaboration.
- This is a limited summary, but includes references.

Contents

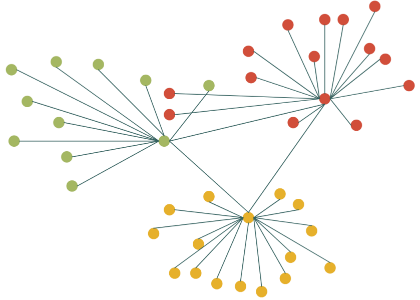


The problem, and why it seems hard



- Across the life cycles manufactured products, computer programs, information artifacts, and other planned, engineered, scientific, or otherwise developed products . . .
- Technical communities have established formal “frameworks for checking” that various intermediate-stage artifacts created during early and later life cycle stages are “consistent” with each other or with various externalities. [6] [7] [10]
- Examples:
  - Is the performance of an engineered product consistent with the specified product technical requirements?
  - Are the predictions a scientific model consistent with the real phenomenon it describes?
  - Are the capabilities of a generated product consistent with the product user’s intended utilization of it?
  - Is the actual in-service use and maintenance of a system consistent with what its specification assumed?
  - Are the specified requirements for a consumer product consistent with what the product’s stakeholders want or need?
  - Are components being fabricated by a contractor consistent with design specified by the integrator purchasing them?
  - Are the tests performed by a component or subsystem supplier consistent with the specifications of the integrator?
  - Is the plan for testing a subsystem consistent with the specification for that subsystem?
  - Are the analyzed risks of a specific use for a product consistent with the understanding of those who are at risk?
  - Many other types of consistencies
- In practice, these are not called “consistencies”--they have individual specific names . . .

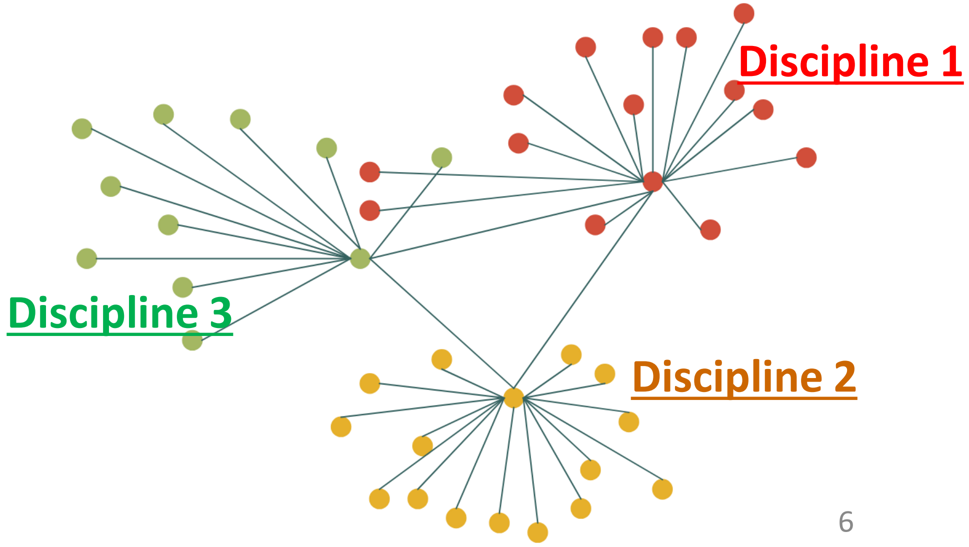
The problem, and why it seems hard



- Examples of more specific consistencies names—for different communities of practice:
  - Computational modeling community: An implemented computational model is *verified* by comparing its output to the conceptual model that guided its implementation. A computational model is *validated* by comparing its predictions to the real system it simulates.
  - Systems engineering community: System requirements are *validated* as to their consistency with the stakeholder needs and requirements they support. An implemented system is *verified* as to it satisfying the requirements for that system. An implemented system is *validated* as to it satisfying the stakeholders.
  - Acquisition community: A newly developed system is subject to *acceptance testing* to determine its satisfaction of system requirements. Incoming purchased parts and materials are subject to *incoming inspection* to release them into product integration.
  - A program-specific supply chain community: Acme Parts Fabrication Corp. performs *product quality inspection* on parts it produces for Quality System Integrators, Inc., which applies *part testing*.
- These and other examples, the result of decades of practice and experience, are formally described by industry and international consensus standards, from standards bodies and technical societies, and regularly updated. [6] [7] [10]
- They are also the subject of extensive company-specific policies and procedures.
- They “hold together” the integrity of the work products of our technical world.
- So what is the problem?

So what is the problem?

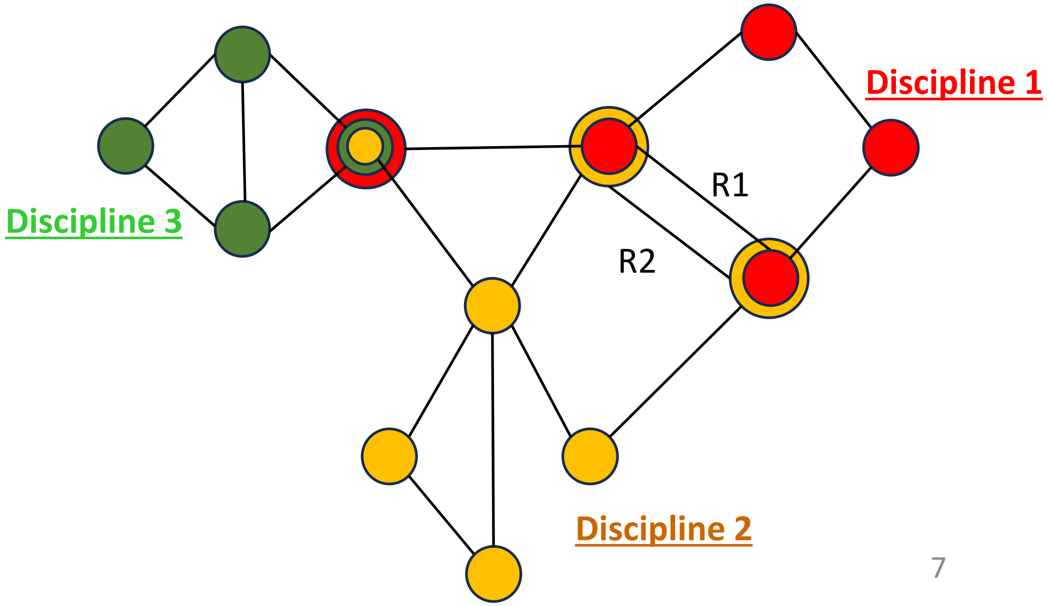
- The communities of practice using these different frameworks are not isolated from each other.
- The ultimate things they create – manufactured products, computational models, computer programs, information products, etc.– cause these communities' roles to not just connect, but overlap.
- The overlap is because the different disciplines/organizations must be able to refer to what they are exchanging—so the artifacts they exchange are the subject of the "semantics" of those interfaces.
- A diagram like this understates the problem:



Some artifacts are *common to different disciplines*, so their disciplines need to refer to them and related processes– but they have *different names* in those disciplines. In other cases, they use the *same name* to refer to *different* things:

- **Problem Example**: A new production parts machining system is being developed to include an embedded *computational model* to predict tool wear, based on duty cycle and raw material types. The *systems engineering* organization has allocated certain *requirements* to the computational model and other requirements to the machine tool, control system, and operator, in its overall integrated system design. The computational modeling organization does not use the term "requirements" for what the computational model should do, but has trustworthy methods for *validating* that a computational model is fit for such use. The systems engineering department has established methods for *validating* the requirements it allocates to the computational model, and for *validating* the machine tool after its integration with the computational model. The computational modeling department has established methods for *verifying* that the implemented computational model is consistent enough with the conceptual model of machine tool wear, for this application. The systems engineering department has established methods for *verifying* that the integrated machine tool meets requirements. During the development process, some of these processes for checking may signal problems not yet solved, and later they may show acceptability. How shall the two disciplines communicate with each other and manage things effectively during development?

The facts that (1) we have different names for the same things, (2) the same names for different things, and (3) formal standards procedures, and learned disciplines that firmly entrench them across large communities may seem to make this a "hard" problem across the disciplines.

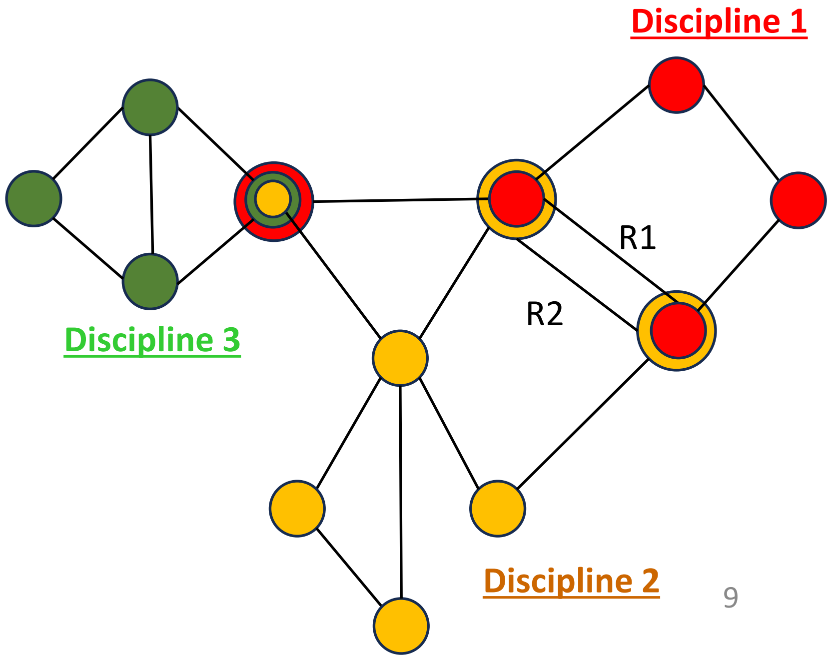


Why this problem must be addressed

- Smart professional teams have been “working around” these issues—so why do we need to do anything different than the status quo?
- Because:
  - Increasingly large, complex and safety-critical systems are being created every year.
  - The need is growing to formalize trust in the integrity of the systems we create and depend on.
  - The ability to move faster in system development and update is a funded imperative demanded by defense establishments.
  - It would be irresponsible to wait for disasters to occur before acting.
  - All these disciplines are highly accomplished, but we hear corrosive disrespect expressed for each other because of different frameworks.
  - The rise of the digital thread, such as described by AIAA and INCOSE thread reference models [1], demands a formal understanding of the semantics of these frameworks.
  - Safe and effective introduction of machine learning systems likewise presses for a solid understanding.
- While the above helps make the case for action, it does not tell us a solution.

Consistency management as a bridging framework for understanding

- “Consistency management” is a description of the core need that can make the problem easier and suggests practical solutions for only modest effort. [3] [11]
- There does not seem to be significant disagreement that disciplined checks for consistency are essential for trustable systems.
- We have different naming schemes for the artifacts being checked, represented by the colors of the nodes in the diagram.
- The consistency checks themselves are represented by the lines (links) between the nodes:
  - In the diagram, R1 and R2 represent differently named checks for substantially the same consistency relationships between differently named (colored) nodes.
  - *<u>We don’t have to change any names to have a “mapping”.</u>*
  - Appendix A provides a simple example

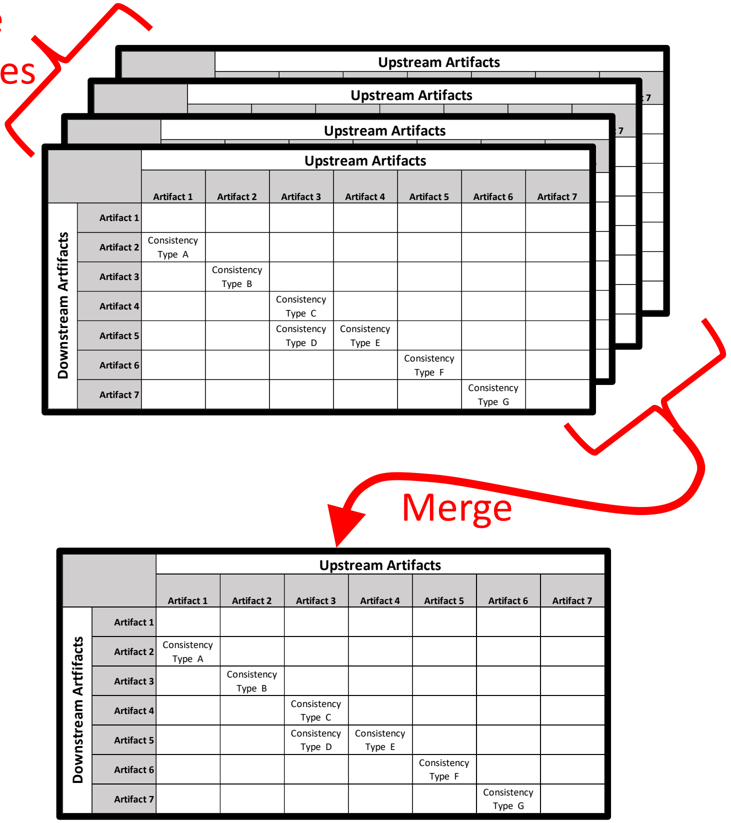


For each discipline (e.g., computational modeling, systems engineering, etc.), an N^2 artifact matrix (a form of "adjacency matrix" for the related graph) of artifact types can be used to display (in the center cells) what consistency check types apply in that discipline, and what they are called:

		Upstream Artifacts						
		Artifact 1	**Artifact 2**	**Artifact 3**	**Artifact 4**	**Artifact 5**	**Artifact 6**	**Artifact 7**
Downstream Artfifacts	**Artifact 1**							
	Artifact 2	Consistency Type A						
	Artifact 3		Consistency Type B					
	Artifact 4			Consistency Type C				
	Artifact 5			Consistency Type D	Consistency Type E			
	Artifact 6					Consistency Type F		
	Artifact 7						Consistency Type G	

For one discipline

Multiple disciplines



Merge

Merged multiple discipline mapping

- Multiple matrices for same artifacts provide "Rosetta Stone" mapping on inter-disciplinary consistency checks.
- Related to Credibility Assessment Frameworks (CAFs) [12].
- See Appendix A for examples.

Recommended social strategy for a cross-society conversation

- Not so hard: This approach does not require any changes to discipline nomenclatures!
- It simply captures them in a common, shared (matrix) representation(s) that makes their coverages and relationships evident.
- So, it need not be extremely difficult.
- But it does imply a conversation between the disciplines.
- Accordingly, the recommended “social strategy” is to carry this out as a collaboration between the technical societies associated with the disciplines.
- For example: ASME, NAFEMS, INCOSE, AIAA, others.

Broader related work already underway in the technical societies

- AIAA/INCOSE/NAFEMS collaboration on Digital Thread reference model [1] [9]: Emphasizes consistency management as the digital thread holds the history of the artifacts and their consistencies.
- ASME VV50 guideline: On the interaction of the model life cycle with the management of model VVUQ, in advanced manufacturing et al. [8]
- INCOSE Innovation Ecosystem (ASELCM) Pattern: Patterns Working Group collaborations with the above groups and others, providing a model-base representation of consistency management as the core of the life cycle. [2]

Discussion and next steps

-
-
-
-
-
-

References

1. AIAA Digital Thread and Digital Twin Reference Models, from INCOSE ASELCM Pattern: Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:aiaa_reference_models_2023.pdf
2. “Planning, Implementing, and Evolving the Ecosystem: Realizing the Promise of Digital Engineering” in *Proc of INCOSE 2022 International Symposium*, Detroit, MI. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:incose_is2022_realizing_the_promise_of_digital_engineering_v1.1.3.pdf
3. “Consistency Management as an Integrating Paradigm for Digital Life Cycle Management with Learning”. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:aselcm_pattern_--_consistency_management_as_a_digital_life_cycle_management_paradigm_v1.3.1.pdf
4. Oberkampf, W. and Roy, C., “Verification and Validation in Scientific Computing”, Cambridge U. Press, 2010.
5. Schlesinger, S., "Terminology for Model Credibility", *Simulation*, 32(3), 103-104, 1979.
6. ISO, "ISO/IEC/IEEE 15288-2023: ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes", 2023.
7. ASME V&V 10-2019: Standard for Verification and Validation in Computational Solid Mechanics. (2019). ASME.
8. Hightower, et al, "Verification and Validation Interactions with the Model Life Cycle: Status of a VV50 Working Group", Proc of ASME V&V Symposium, May 2021. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:model_life_cycle_working_group_status_v1.2.5.pdf
9. Taylor, N., and Schindel, W., “CFD Validation: Illustrations of Mutual Accountability and Validation Dialog throughout the Engineering Lifecycle”, paper to be presented at AIAA 2024 SciTech Conference, Orlando, FL.
10. Walden, D., et al, eds., “INCOSE Systems Engineering Handbook”, Fifth Edition, International Council on Systems Engineering, San Diego, Ca, 2023.
11. “All Decisions Across Life Cycles of Systems Are Reconciliations of Inconsistencies”, INCOSE North Texas Program, August, 2023. Download from --_https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:incose_north_texas_pgm_08.08.2023_v1.2.2.pdf
12. Kaiser, J. (2019). Credibility Assessment Frameworks for Empirical/Data Driven Models – Personal Views. ASME V&V Symposium. Las Vegas: ASME.

Related author technical society activities

- The author is a member of INCOSE, ASME, AIAA, and ASEE, with historical publications in all four.
- INCOSE: Founding chair of INCOSE Patterns Working Group; INCOSE Fellow.
- ASME: Active member of ASME VV50 working group on advanced manufacturing models credibility management across the model life cycle.
- AIAA: Member of authoring teams for AIAA reference models for Aerospace Digital Threads and Aerospace Digital Twins.
- ASEE: Publications on systems engineering education for undergraduates.

Appendix: A simple example of "mapping" without any changes in nomenclature

- Computational Modeling V&V
- Systems Engineering V&V

One perspective from the computational modeling community

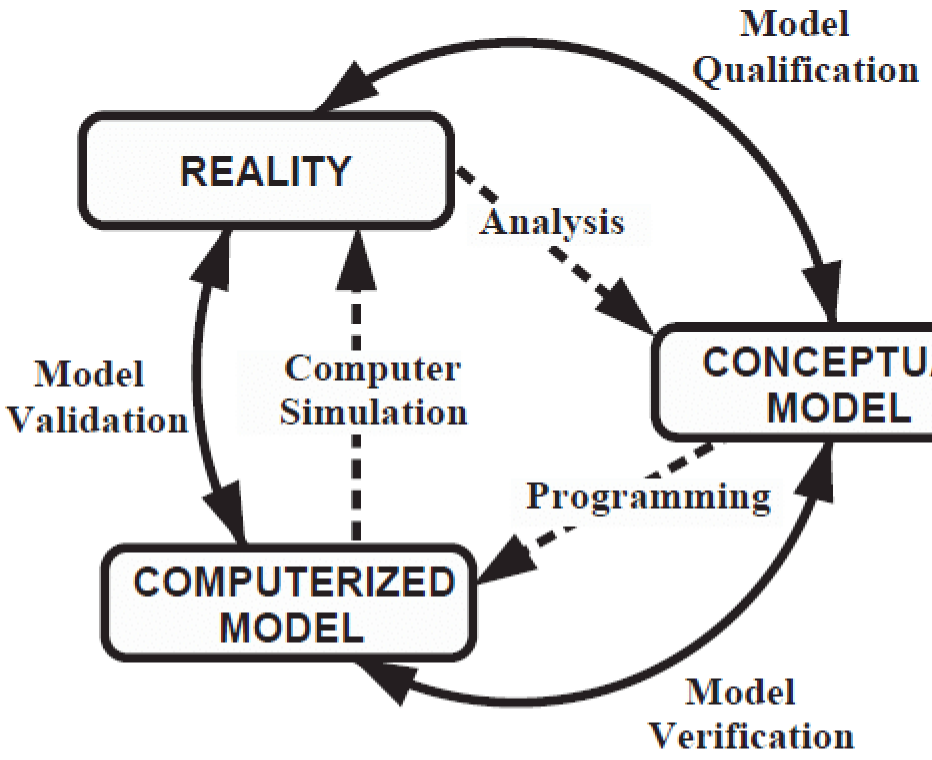
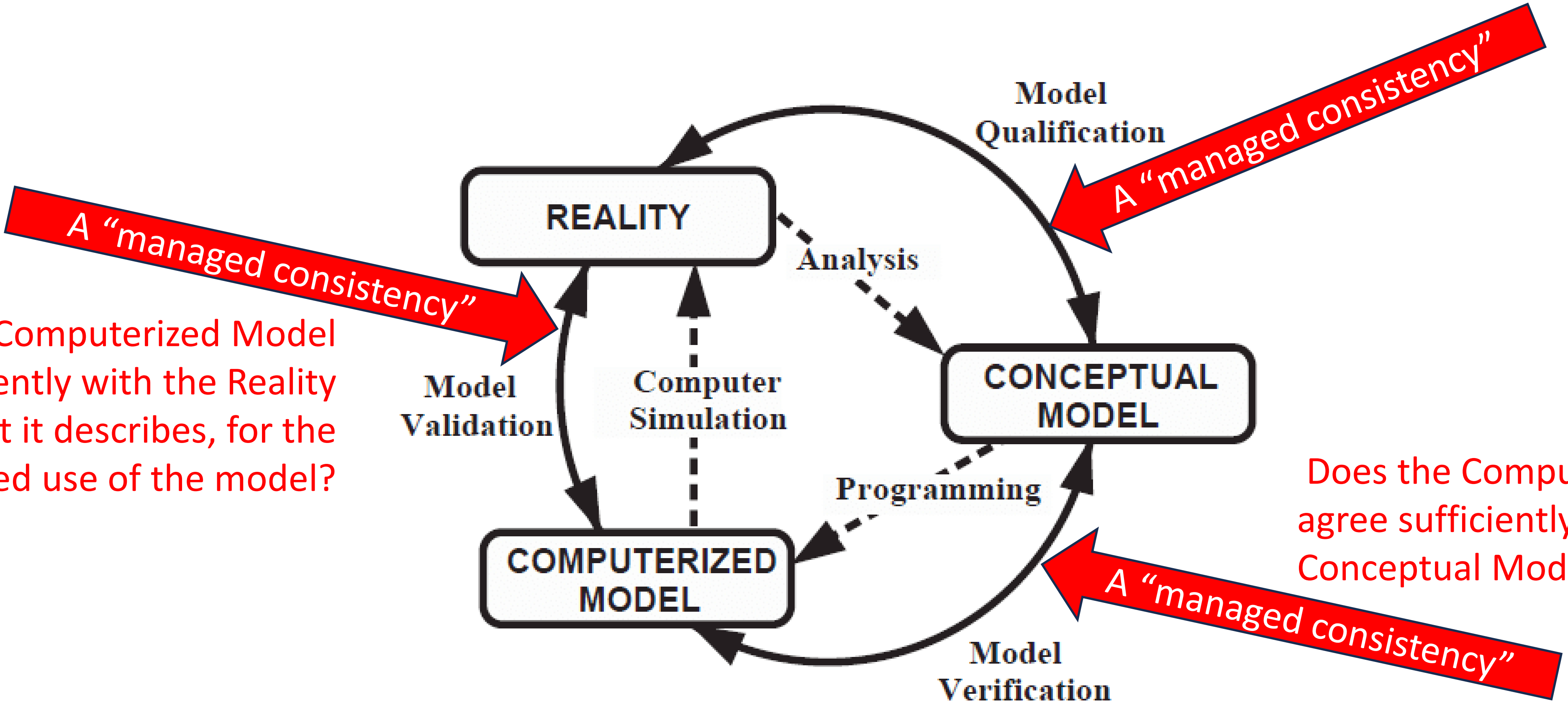


Diagram: The role of V&V in the development of simulation models [(Schlesinger, 1979) [5].

- This diagram is somewhat dated by subsequent developments, but offers a simple example using ideas that continue to apply.
- An informative discussion of this diagram and subsequent history is in Oberkampf and Roy (2010) [4], pp 22 and its following sections.

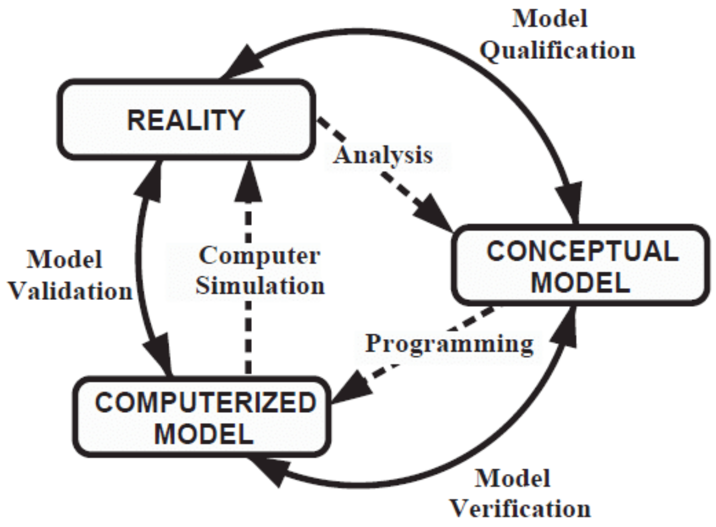
In the language of “managed consistencies”

Is the Conceptual Model suitable as a model of Reality, for the intended purpose?



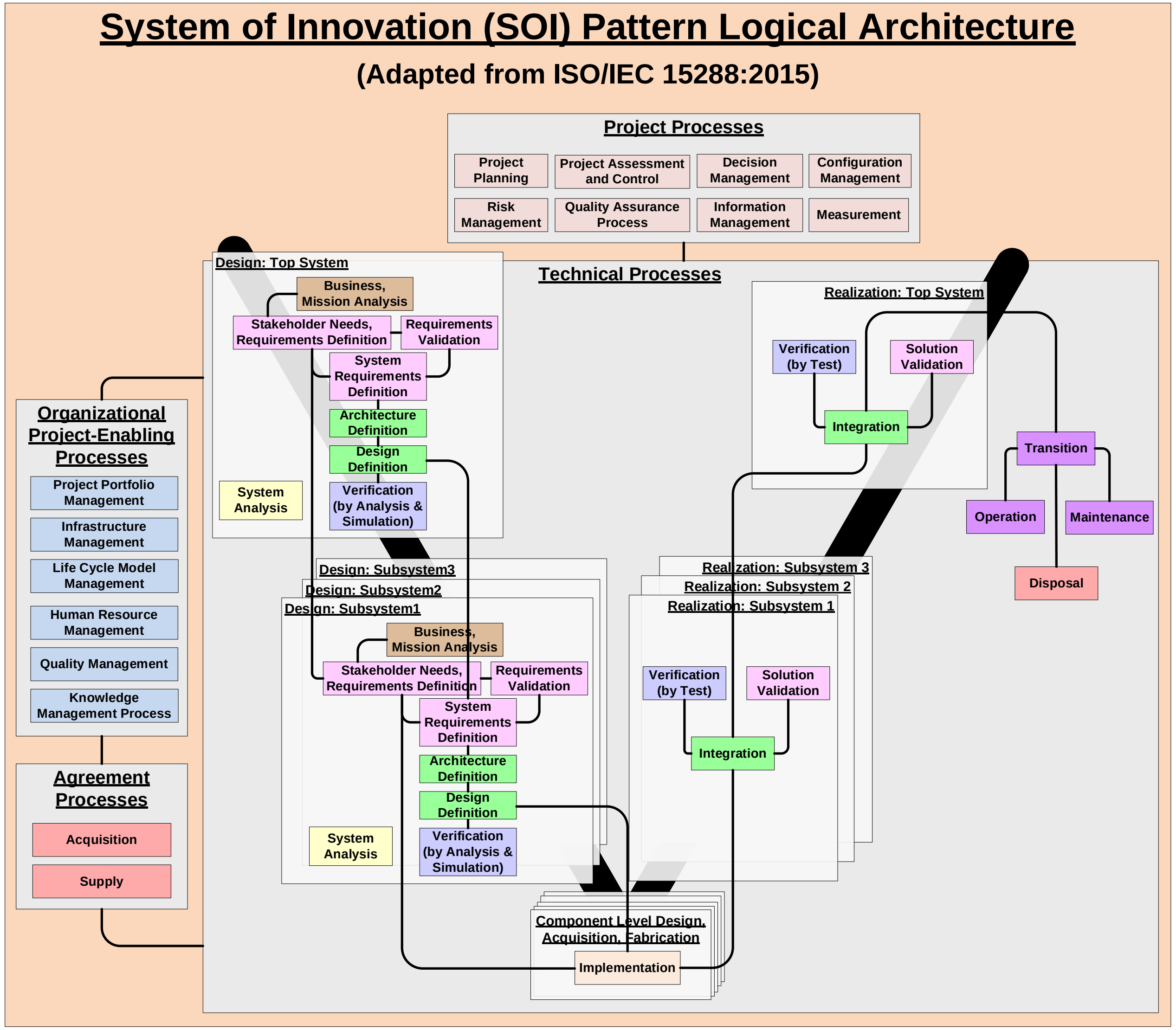
Does the Computerized Model agree sufficiently with the Reality system that it describes, for the intended use of the model?

Does the Computerized Model agree sufficiently with the Conceptual Model it is based upon?



		Upstream Artifacts						
		Model Stakeholders	**Real Modeled System**	**Computatinal Model Requirements**	**Conceptual Model**	**Model Non-Accuracy Requirements**	**Computational Model Design**	**Realized Computational Model System**
Downstream Artifacts	**Model Stakeholders**							
	Real Modeled System							
	Computational Model Requirements							
	Conceptual Model		Conceptual Model Qualification					
	Model Non-Accuracy Requirements							
	Computational Model Design							
	Realized Computational Model System		Model Validation (Accuracy)		Model Verification (Accuracy)			

Systems Engineering "Vee" Perspective on Engineering of Systems (as in ISO15288 [6], INCOSE Handbook [10], etc.)



System of Innovation (SOI) Pattern Logical Architecture

(Adapted from ISO/IEC 15288:2015)

Project Processes

Project Planning
Project Assessment and Control
Decision Management
Configuration Management
Risk Management
Quality Assurance Process
Information Management
Measurement

Technical Processes

Design: Top System
Business, Mission Analysis
Stakeholder Needs, Requirements Definition
Requirements Validation
System Requirements Definition
Architecture Definition
Design Definition
Verification (by Analysis & Simulation)
System Analysis

Realization: Top System
Verification (by Test)
Solution Validation
Integration

Transition
Operation
Maintenance
Disposal

Design: Subsystem3
Design: Subsystem2
Design: Subsystem1
Business, Mission Analysis
Stakeholder Needs, Requirements Definition
Requirements Validation
System Requirements Definition
Architecture Definition
Design Definition
Verification (by Analysis & Simulation)
System Analysis

Realization: Subsystem 3
Realization: Subsystem 2
Realization: Subsystem 1
Verification (by Test)
Solution Validation
Integration

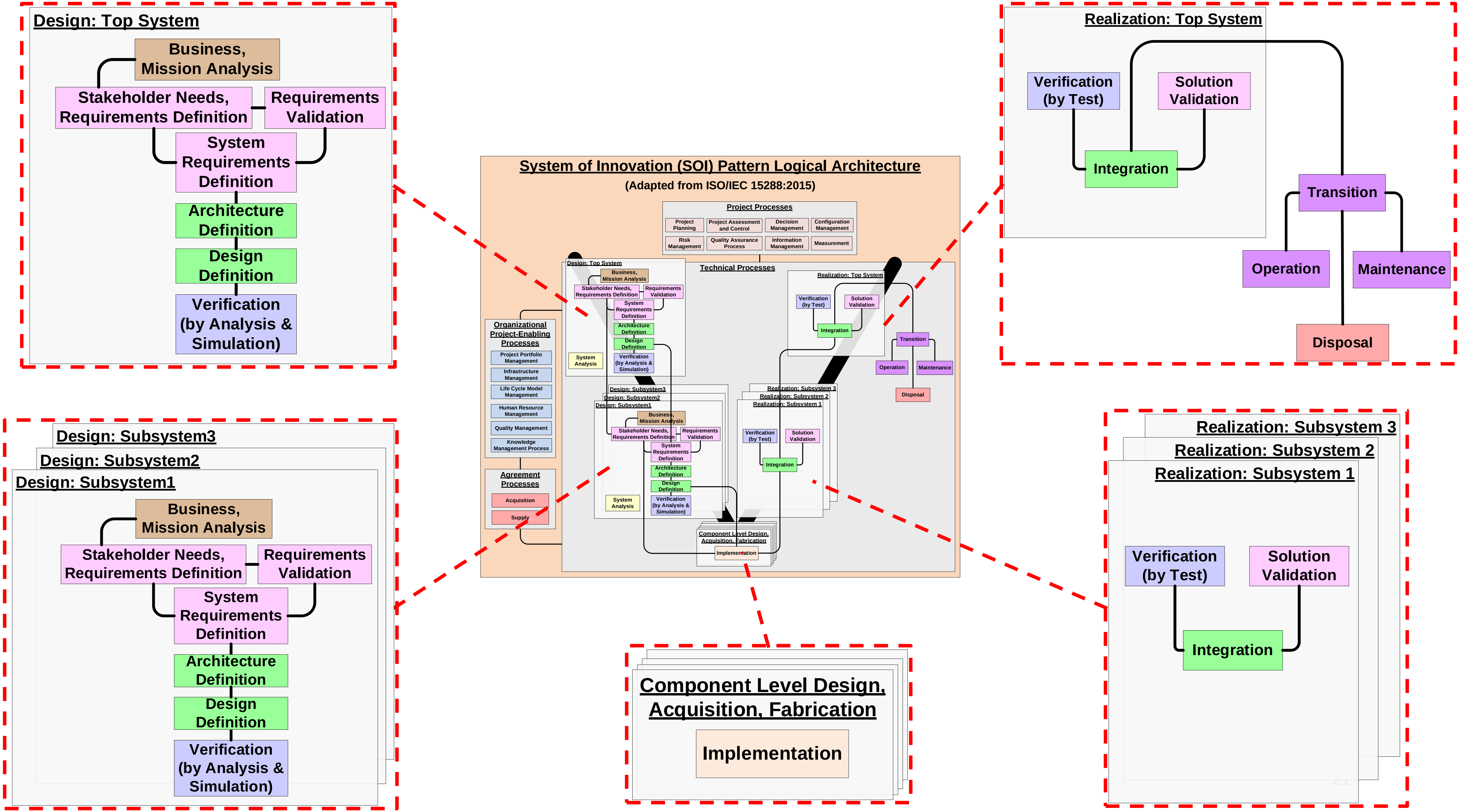
Component Level Design, Acquisition, Fabrication
Implementation

Organizational Project-Enabling Processes

Project Portfolio Management
Infrastructure Management
Life Cycle Model Management
Human Resource Management
Quality Management
Knowledge Management Process

Agreement Processes

Acquisition
Supply



System of Innovation (SOI) Pattern Logical Architecture

(Adapted from ISO/IEC 15288:2015)

Project Processes

Project Planning
Project Assessment and Control
Decision Management
Configuration Management
Risk Management
Quality Assurance Process
Information Management
Measurement

Organizational Project-Enabling Processes

Project Portfolio Management
Infrastructure Management
Life Cycle Model Management
Human Resource Management
Quality Management
Knowledge Management Process

Agreement Processes

Acquisition
Supply

Technical Processes

Design: Top System

Business, Mission Analysis
Stakeholder Needs, Requirements Definition
Requirements Validation
System Requirements Definition
Architecture Definition
Design Definition
Verification (by Analysis & Simulation)
System Analysis

Design: Subsystem3

Design: Subsystem2

Design: Subsystem1

Business, Mission Analysis
Stakeholder Needs, Requirements Definition
Requirements Validation
System Requirements Definition
Architecture Definition
Design Definition
Verification (by Analysis & Simulation)
System Analysis

Component Level Design, Acquisition, Fabrication

Implementation

Realization: Subsystem 3

Realization: Subsystem 2

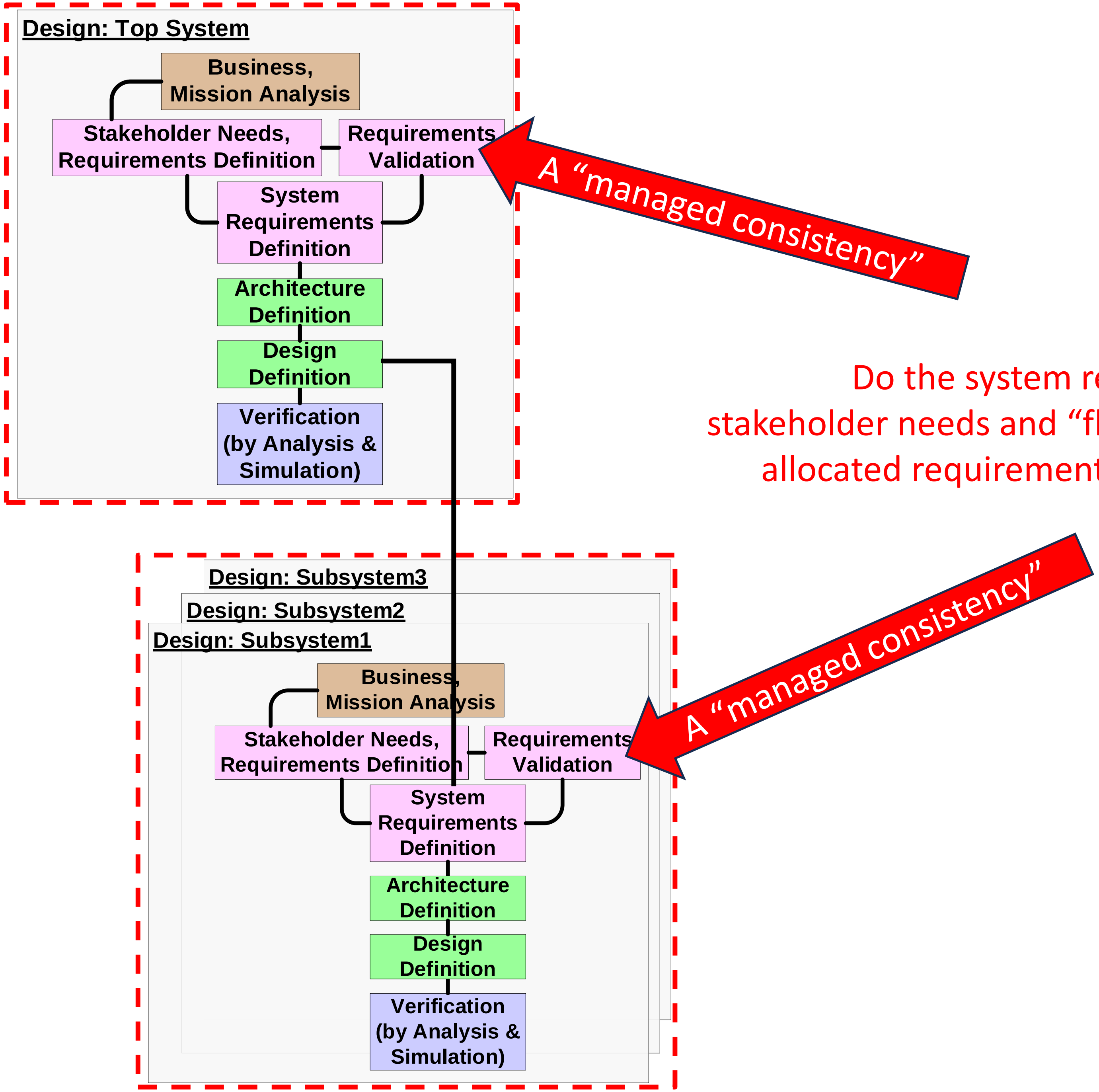
Realization: Subsystem 1

Verification (by Test)
Solution Validation
Integration

Realization: Top System

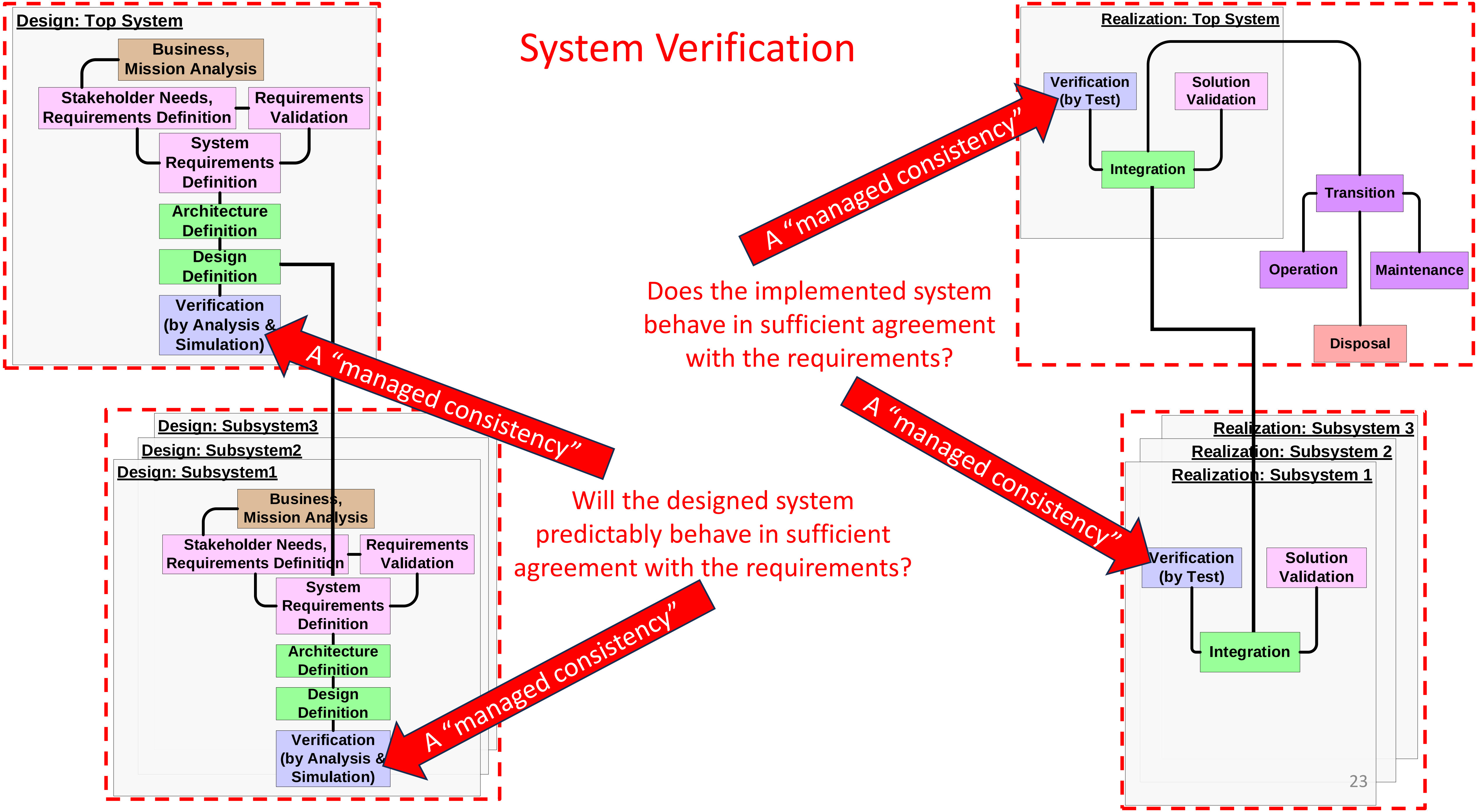
Verification (by Test)
Solution Validation
Integration
Transition
Operation
Maintenance
Disposal

Requirements Validation



Do the system requirements represent the stakeholder needs and “flowed down” decomposed allocated requirements adequately for purpose?

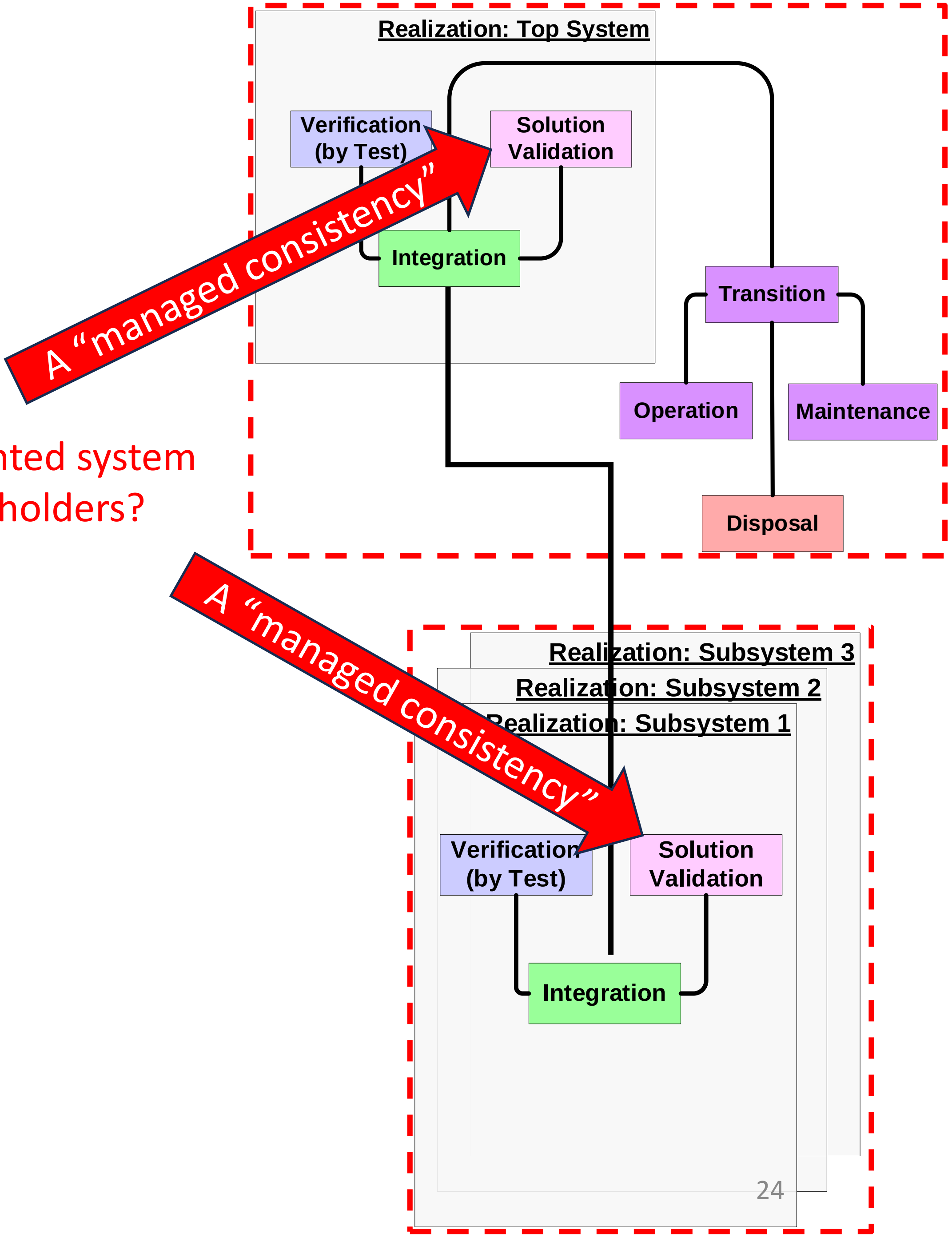
System Verification



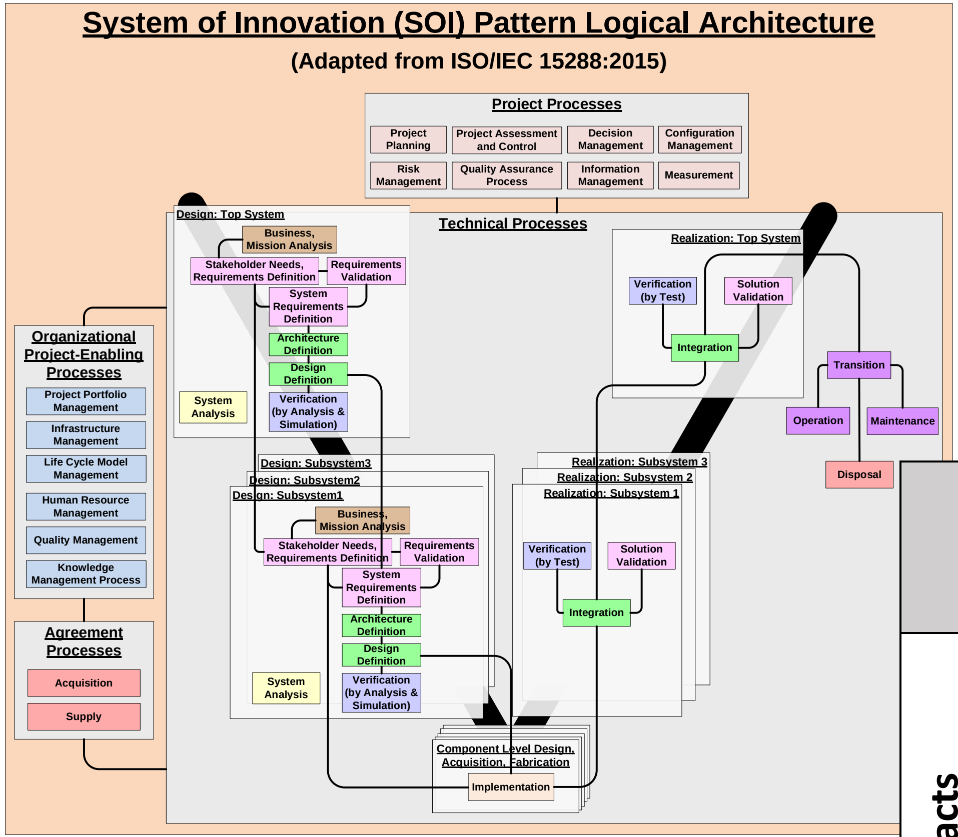
Does the implemented system behave in sufficient agreement with the requirements?

Will the designed system predictably behave in sufficient agreement with the requirements?

Solution Validation



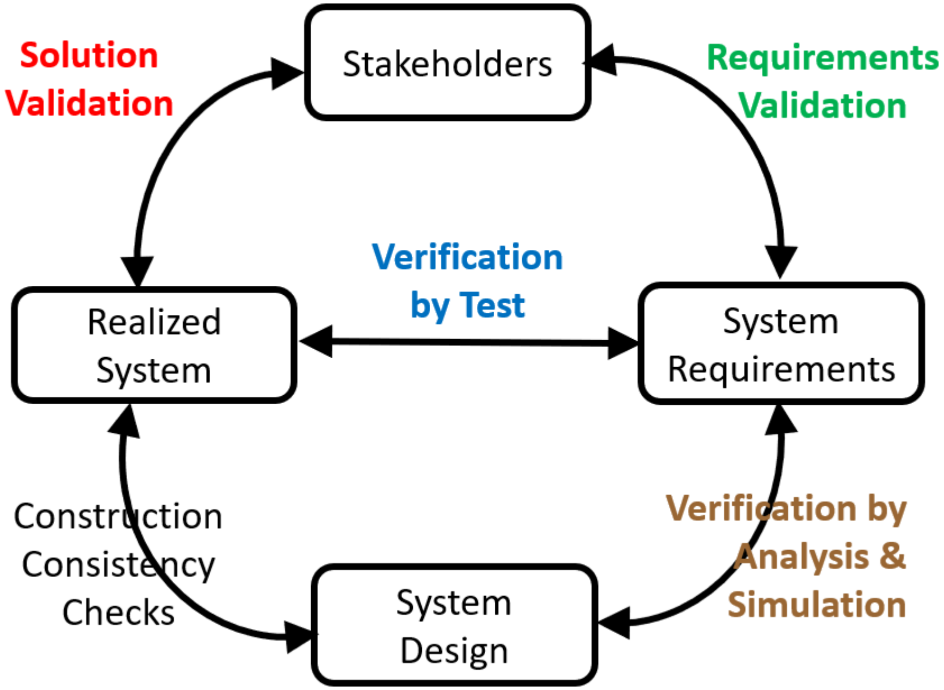
Does the implemented system satisfy the stakeholders?



Downstream Artifacts	Model Stakeholders	Real Modeled System	Computatinal Model Requirements	Conceptual Model	Model Non-Accuracy Requirements	Computational Model Design	Realized Computational Model System
	Upstream Artifacts						
Model Stakeholders							
Real Modeled System							
Computational Model Requirements	Model Requirements Validation						
Conceptual Model							
Model Non-Accuracy Requirements							
Computational Model Design			Model Verification by Analysis				
Realized Computational Model System	Model Validation (Overall)					Construction Consistency Checks	

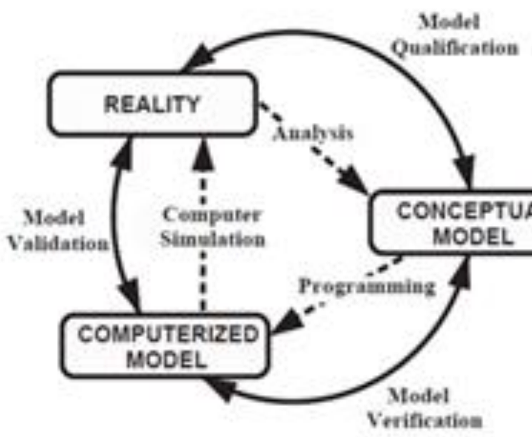
A small subset of the many consistency checks that can apply across the ISO15288 engineered product life cycle

ISO 15288 V&V	**Shown on Left Side of Vee**	**Shown on Right Side of Vee**
Validation: Checking consistency with Stakeholder interests	**Requirements Validation**	**Solution Validation**
Verification: Checking consistency with System Requirements	**Verification by Analyzing, Simulating, inspecting, reviewing planned design against System Requirements**	**Verification by Testing real built or acquired system, subsystems, parts against their System Requirements**

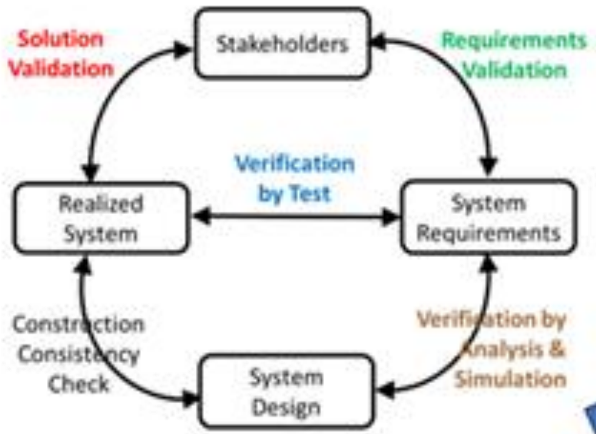


(Vee left and right sides reversed in above circle to ease later alignment with historical computational modeling V&V "circle".)

Historical Computational Model V&V



Historical ISO15288 V&V



Combined V&V View