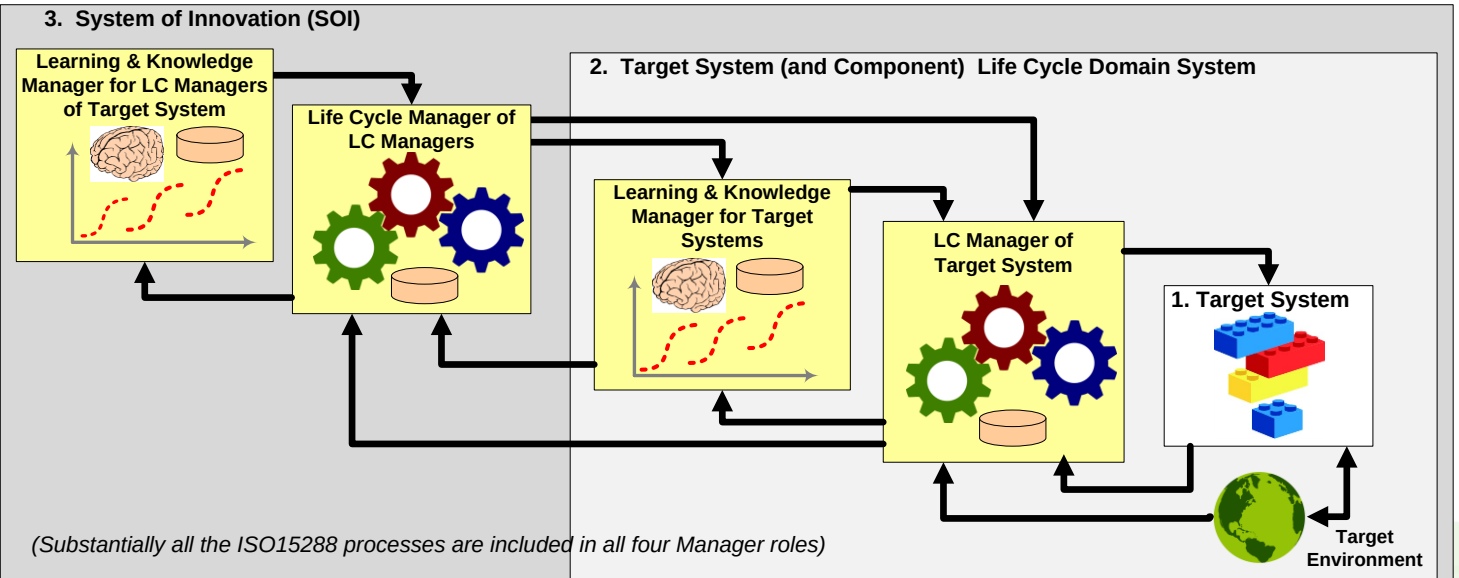
26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern



Bill Schindel
schindel@ictt.com

Rick Dove
rick.dove@parshift.com

Abstract

Engineered and other systems are under pressure to adapt, from opportunities or competition, predators, changing environment, and physical or cyberattack. Ability to adapt well enough as conditions change, especially in presence of uncertainty, is valued. Systems (including developmental and life cycle management) that adapt well enough, in time, cost, and effectiveness, are sometimes called “agile”. As environmental change or uncertainty increase, agility can mean survival.

Agile systems and agile systems engineering are subjects of an INCOSE 2015-16 discovery project, described elsewhere. This paper introduces the underlying MBSE-based Agile Systems Engineering Life Cycle Pattern being used to capture, analyze, and communicate key aspects of systems being studied. More than an ontology, this model helps us understand necessary and sufficient conditions for agility, different approaches to it, and underlying relationships, performance couplings, and principles.

This paper introduces the framework, while specific findings about methods and practicing enterprises studied will be reported separately.

Contents

- What is the INCOSE Agile Systems Engineering Life Cycle Model Discovery Project?
- What are Agile Systems, and why do they matter?
- How are Agile Systems related to MBSE?
- What is the Agile Systems Engineering Life Cycle Pattern?
- Example: Applying the ASELCM Pattern to Plan Agility Improvement
- Where can I learn more?
- Discussion
- References

What is the INCOSE Agile Systems Engineering Life Cycle Model Discovery Project?

- During 2015-16, the INCOSE parent society is sponsoring the Agile Systems Engineering Life Cycle Model (ASELCM) Discovery Project, based on a series of workshop clinics being held at host example discovery sites across the U.S. and Europe.
- This project, now underway, will provide INCOSE inputs to a future version of ISO 15288, to improve explicit understanding of principles and practices of agility as applicable to systems engineering across different domains.

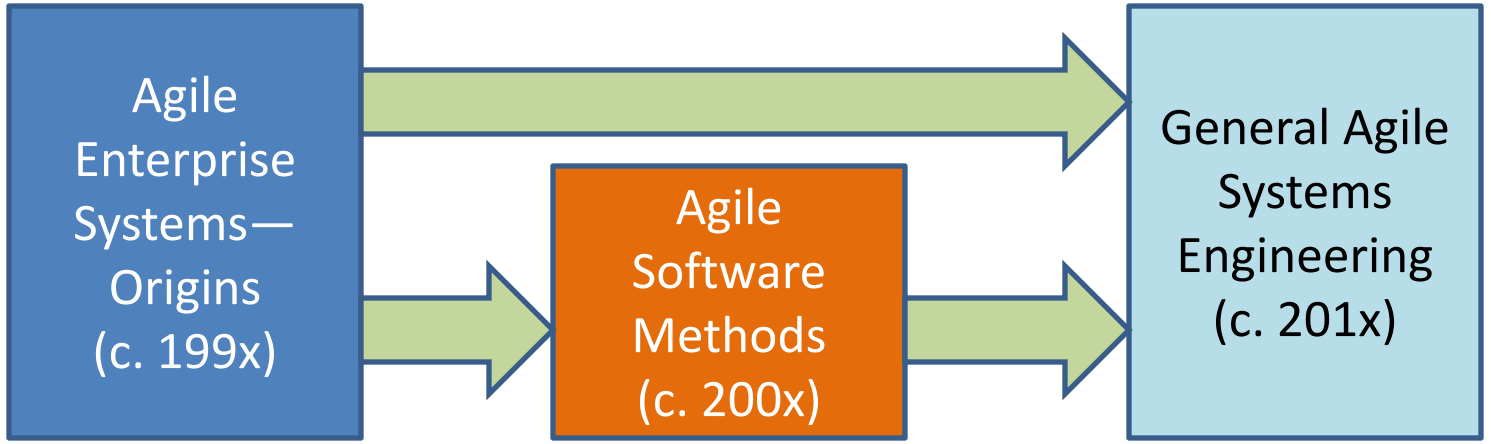
http://www.parshift.com/ASELCM/Home.html

What is the INCOSE Agile Systems Engineering Life Cycle Model Discovery Project?

- Announced at IW2015
- Built around discovery clinics being conducted by example host sites during 2015-16.
- Discovery clinics in 2015:
  - Navy SpaWar/MITRE, San Diego, CA,
  - Northrop Grumman, Vienna, VA,
  - Rockwell Collins, Cedar Rapids, IA,
  - Lockheed Martin, Ft. Worth, TX,
- You and your company can host or participate in 2016!
- Support from INCOSE Agile Systems WG and Patterns WG:
  - R. Dove, project lead, co-leads K.Forsberg, H. Lawson, J. Ring, G. Roedler, B. Schindel

What are Agile Systems? Why do they matter?

Longer history than just Agile Software Development Methods :



- For history and background, see Dove and LaBarge, 2014
- Agile software methods, by far better known, are related.
- General Agile Systems Engineering is the related broader subject of the INCOSE ASELCM Project.
- **Problem space**: *Challenges of uncertainty and rates of change in environment, stakeholders, competition, technologies, capacities, capabilities. Not just "going faster".*

Is this your tomorrow, or a distant vision?

From "The Hardware Renaissance Arrives: A New Dawn for Gadgets", *The Wall Street Journal*, March 23, 2015:

"Recently, as I gazed into the prototype of a smart breast pump, I had a vision of the future. I saw an age in which new products—actual, physical electronics products—will go from idea to store shelves in a matter of months. A future in which warehouses and distribution centers cease to exist, because factories produce finished goods from raw materials on demand, and they never stop moving through the supply chain. *Only it turns out all of this is possible today*. The "hardware renaissance" that began in Silicon Valley in just the last five years, born of rapid prototyping technologies, has become something much larger and more important. It has been a sea change in every stage of producing physical objects, from idea to manufacturing to selling at retail . . ."

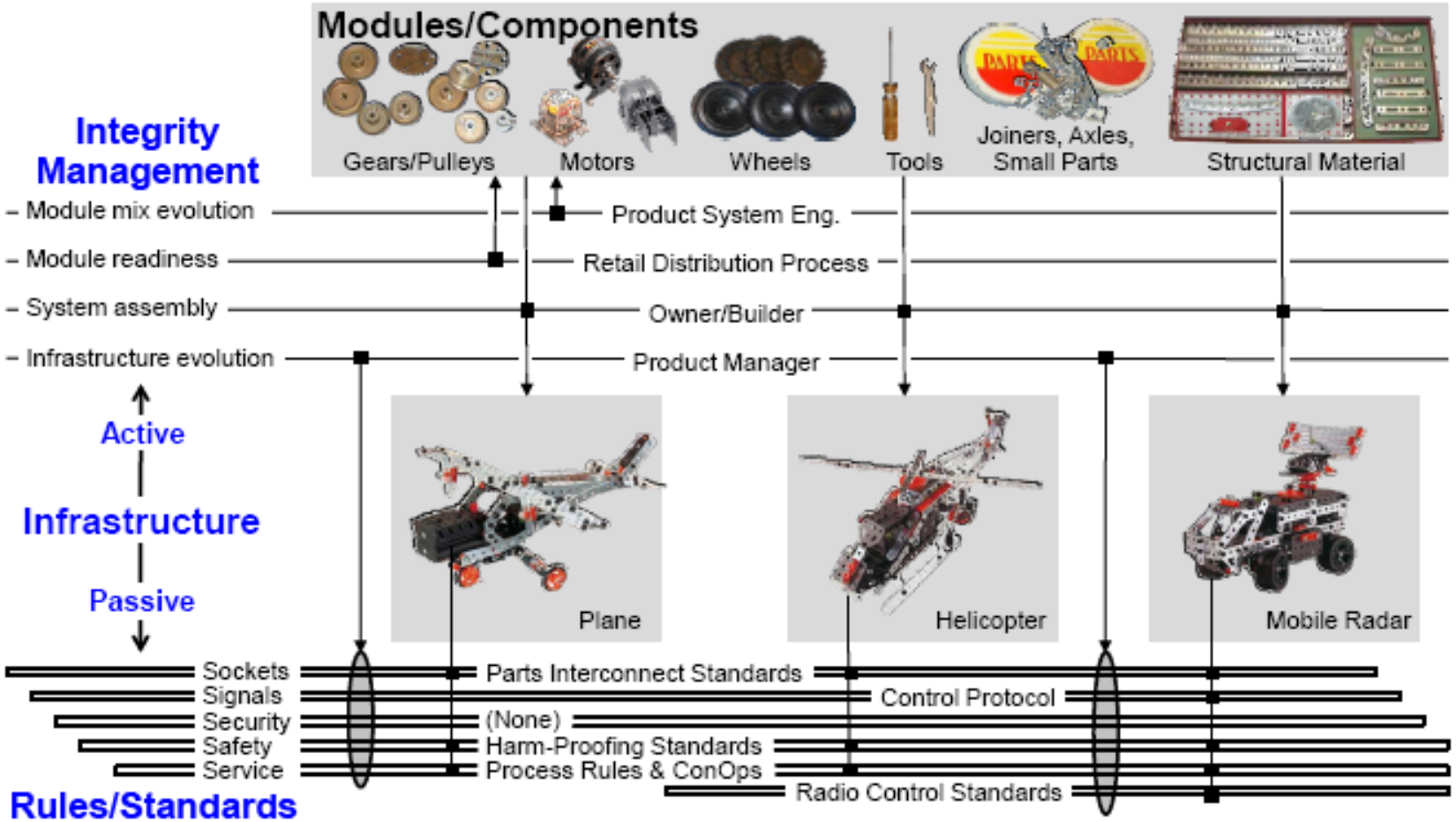
-- Christopher Mims, *The Wall Street Journal*, p B1,6, March 23, 2015

-- emphasis added

Agile Systems Architecture Pattern (R. Dove)

The S*ASELCM Pattern captures (in a formal S*Model) the key ideas associated with the pre-MBSE Agile System Architecture:

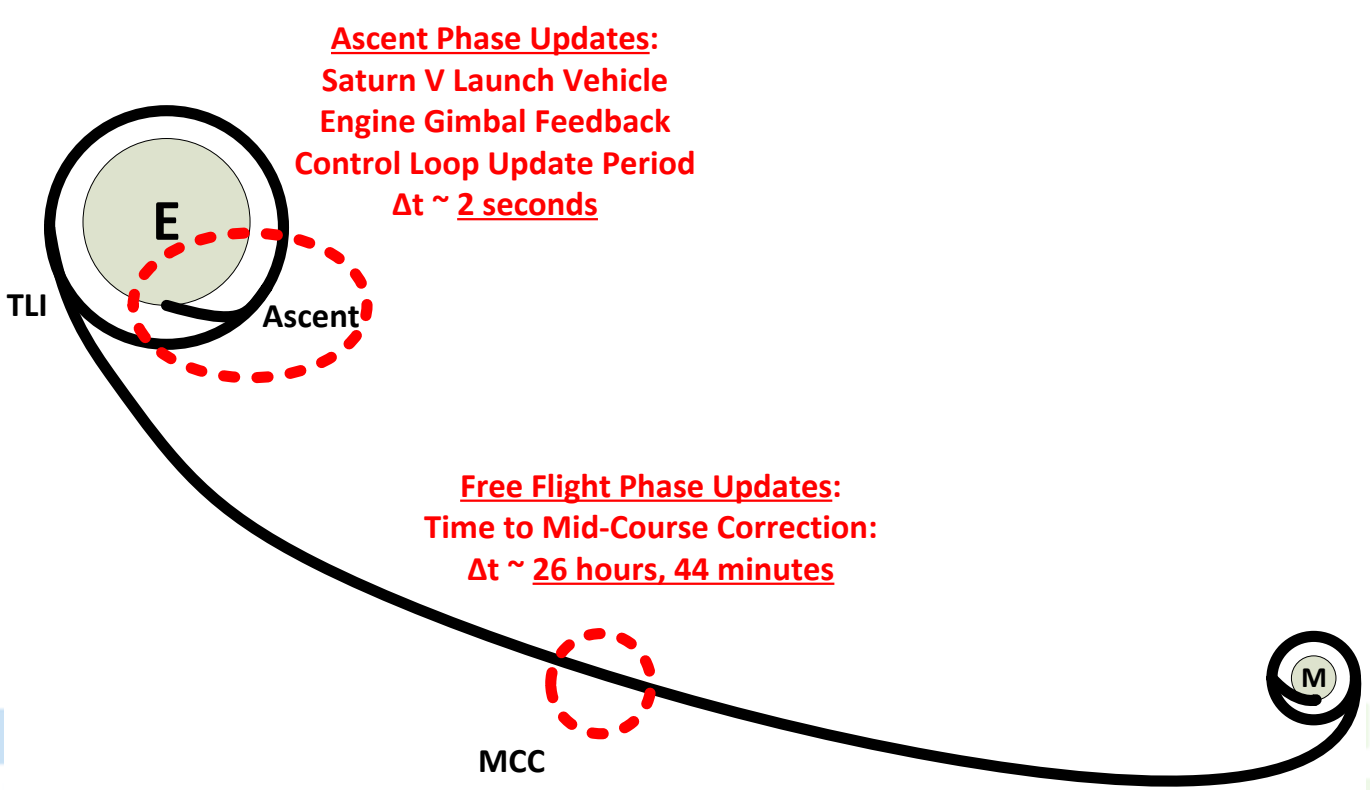
- As in (Dove and LaBarge, 2014)



Optimized Feedback & Correction Cycle Rate:
A Hallmark of Agile Methods & Problem Space

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

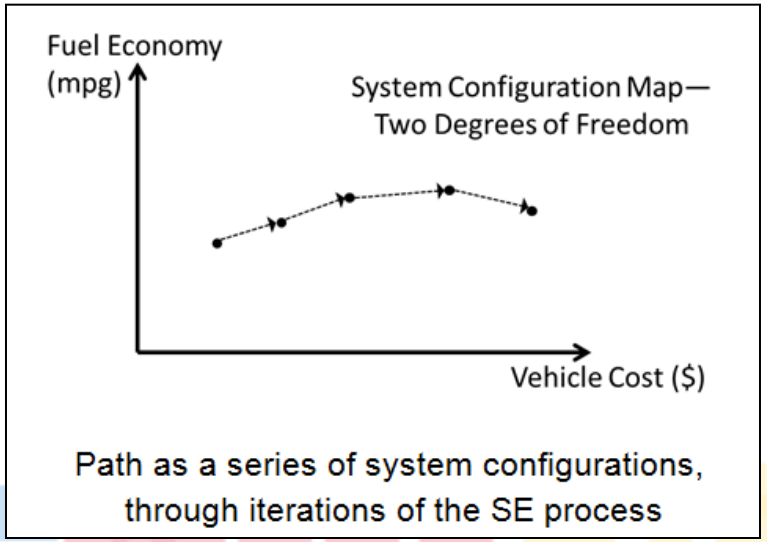
An Apollo 11 Mission Question: Why was the Saturn V rocket engines' directional gimbals update cycle period throughout the Ascent Phase ~ 2 seconds, but the update cycle period of course direction during the Free Flight Phase was ~ 26 hours?



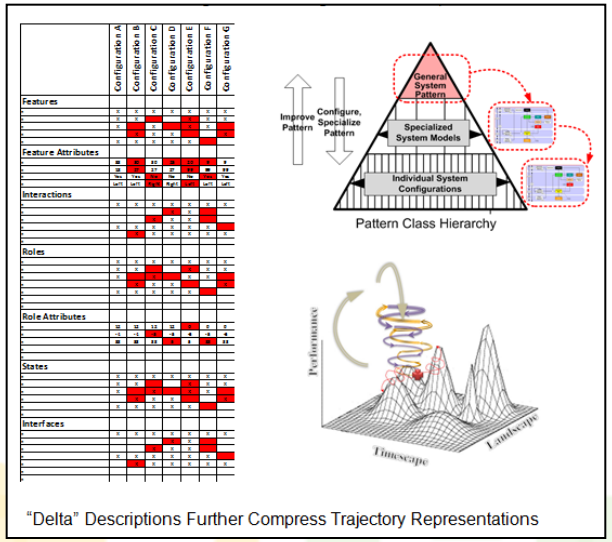
System Life Cycle Trajectories in S*Space

- Configurations change over life cycles, during development and subsequently
- Trajectories (configuration paths) in S*Space
- Effective tracking of trajectories
- History of dynamical paths in science and math
- Differential path representation: compression, equations of motion

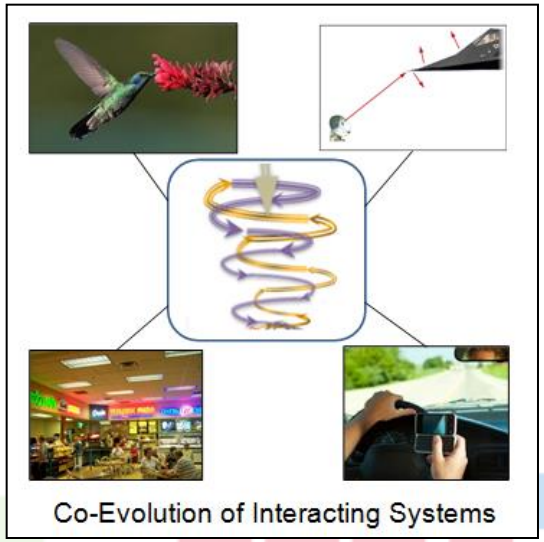
Evolving Systems Over Multiple Life Cycles



Path as a series of system configurations, through iterations of the SE process

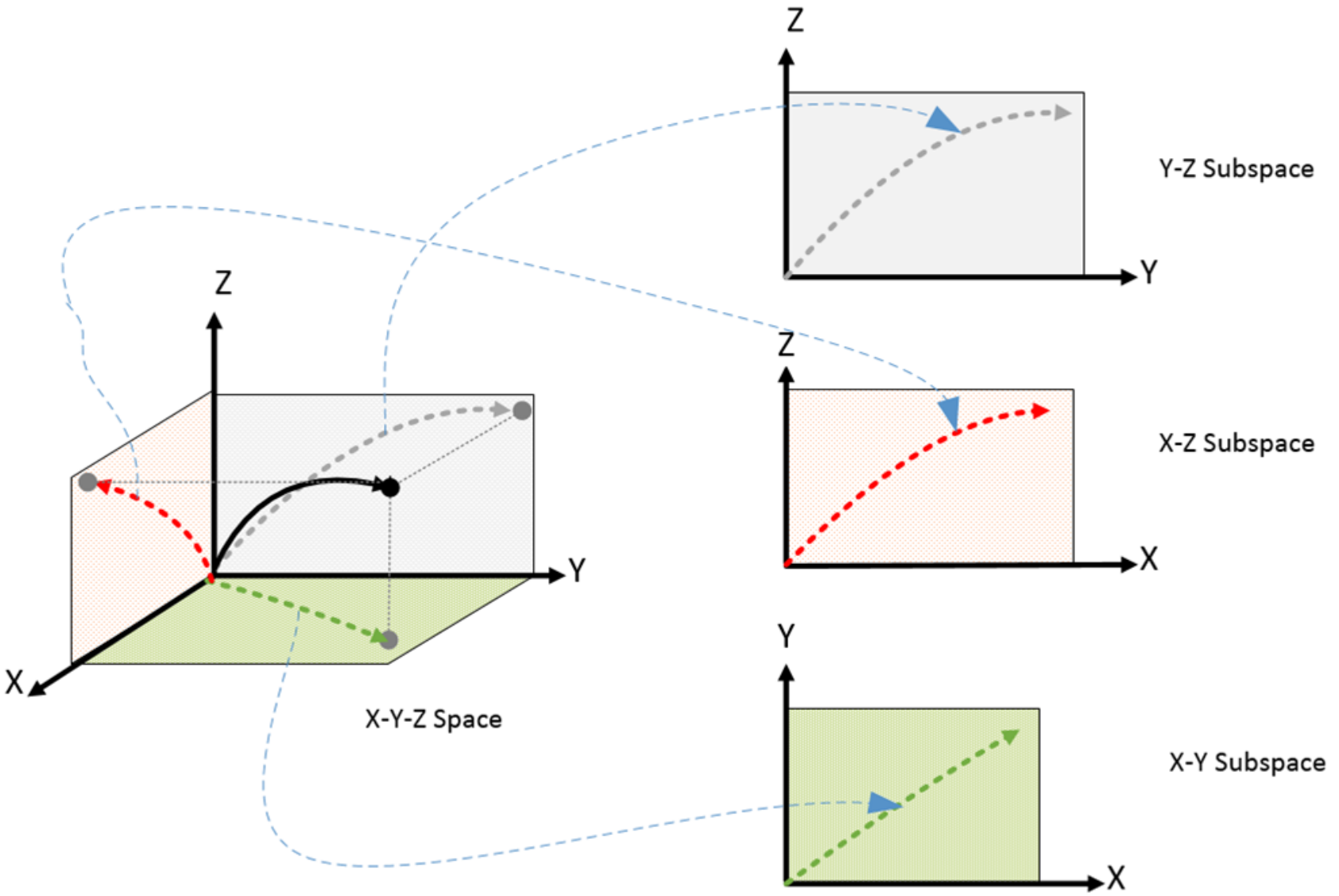


"Delta" Descriptions Further Compress Trajectory Representations

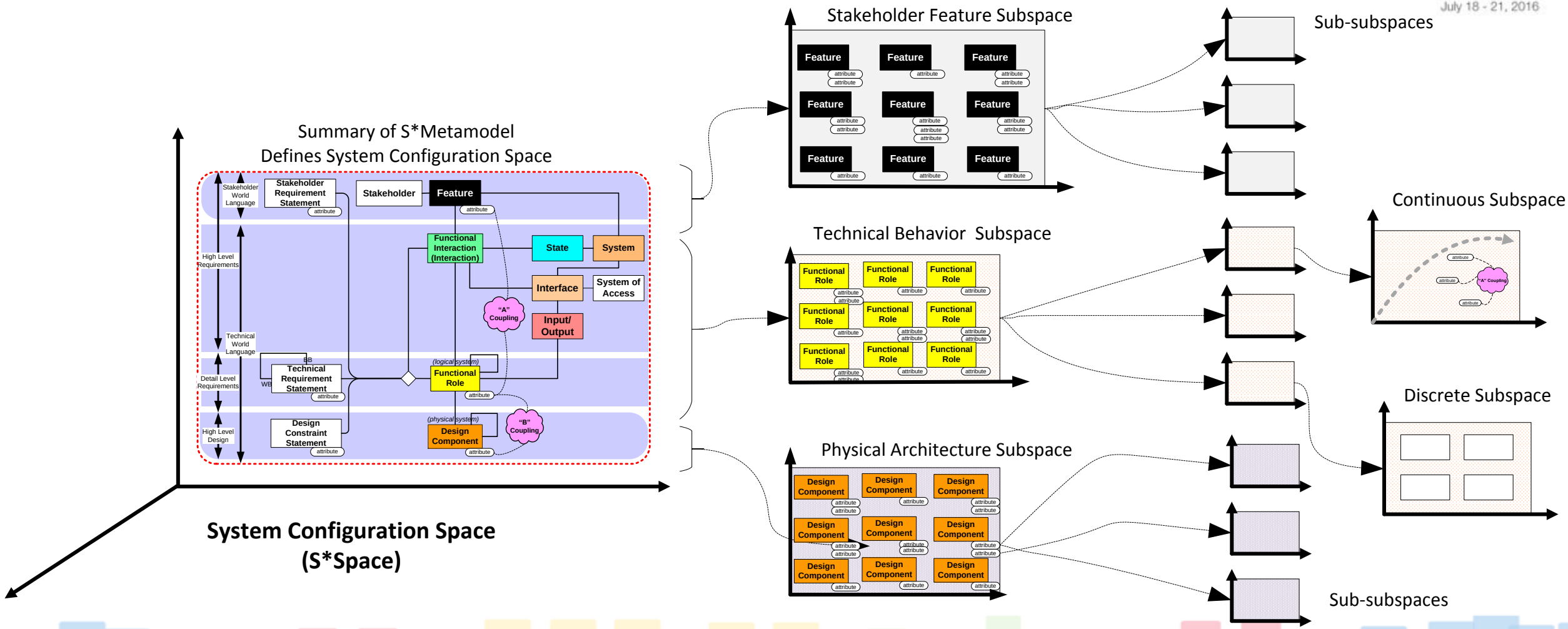


Co-Evolution of Interacting Systems

Simple Geometric/Mathematical Idea: Subspace Projections

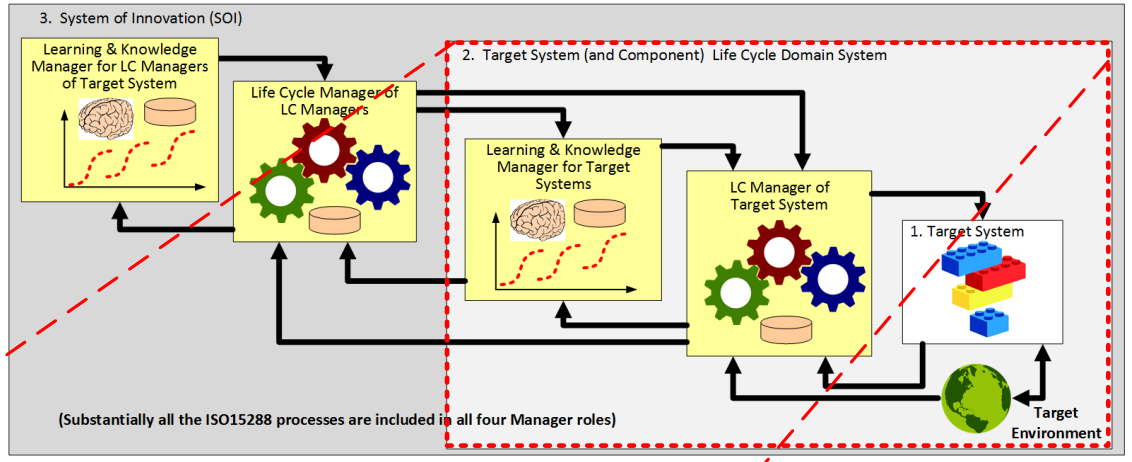


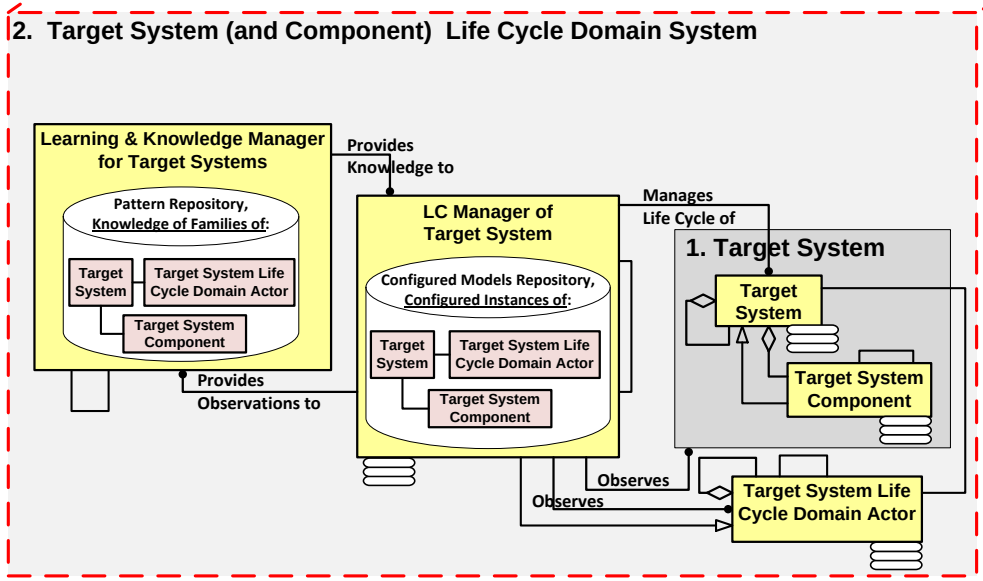
System Life Cycle Trajectories in S*Space, and S*Subspaces

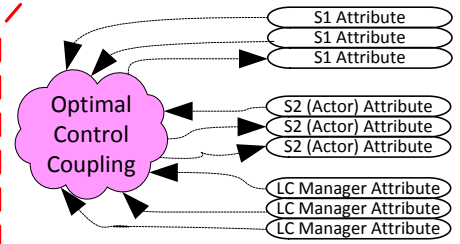


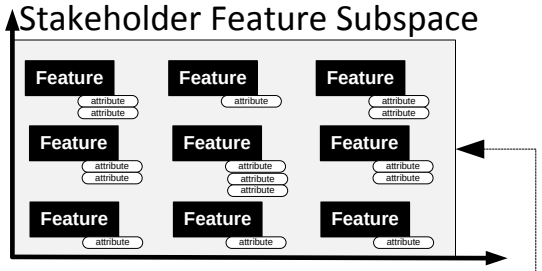
Agility as Optimal Control in S*Space:

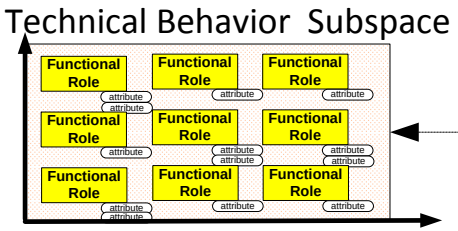
Finding the Best Next Increment "Direction"

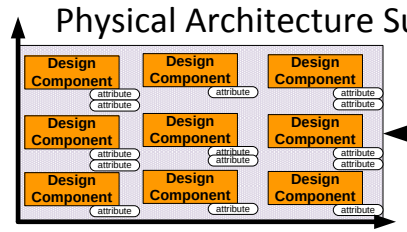


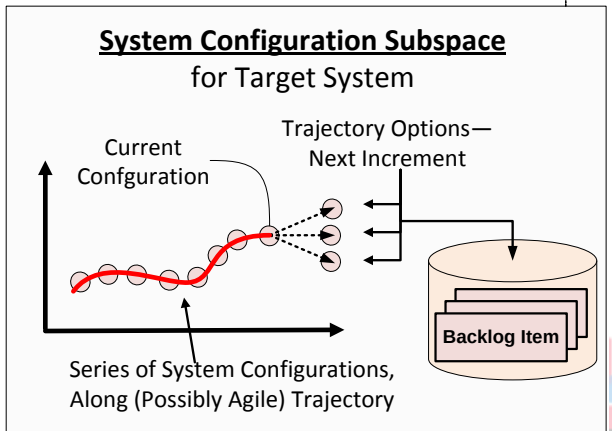












How are Agile Systems Related to MBSE?

1. **Basics**: Using explicit models, MBSE/PBSE adds clarity to pre-model descriptions of Agile Systems and Agile SE-- improves understanding of Agile Systems.
2. **More important**: MBSE/PBSE complements and improves the capability of Agile Systems and Agile Systems Engineering—
   - Agility requires persistent memory & learning—*being forgetful/not learning impacts agility*.
   - Patterns capture & retain learning, as persistent, re-usable, configurable, models, *updated as experience accumulates*.
   - S*Patterns are configurable, reusable S*Models.

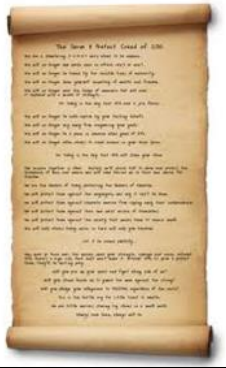
"*PBSE as Agile MBSE*" emerges as essential when competing on agility becomes reality for competing, competent players:

- Improved: "Where are we?"
- Improved: "Where are we going?"
- Improved: "We've been here before."
- Improved: Understanding of response.
- Improved: Understanding of mission envelopes.
- Improved: Ability to assess agility
- Improved: Ability to plan agility

Vital for Scrum, other approaches

Vital for Response Situation Analysis (RSA)

Maps *vs*. Itineraries -- SE Information *vs*. SE Process

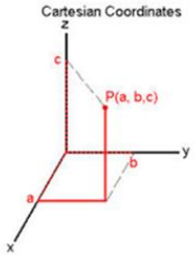


Itinerary ≠ Map!

(What am I doing?) *(Where am I?)*

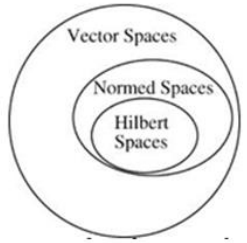
When they eventually did emerge, maps represented a newer idea of the nature of "where".

- The SE Process consumes and produces information.
- But, SE historically emphasizes process over information. (Evidence: Ink & effort spent describing standard process versus standard information.)
- Ever happen?-- Junior staff completes all the process steps, all the boxes are checked, but outcome is not okay.
- Recent discoveries about ancient navigators: Maps *vs*. Itineraries.
- The geometrization of Algebra and Function spaces (Descartes, Hilbert)
- Knowing where you are, not just what you are doing.
- Knowing where you are going, not just what you are doing.
- Distance metrics, inner products, projections, decompositions.



Rene Descartes
1596 - 1650

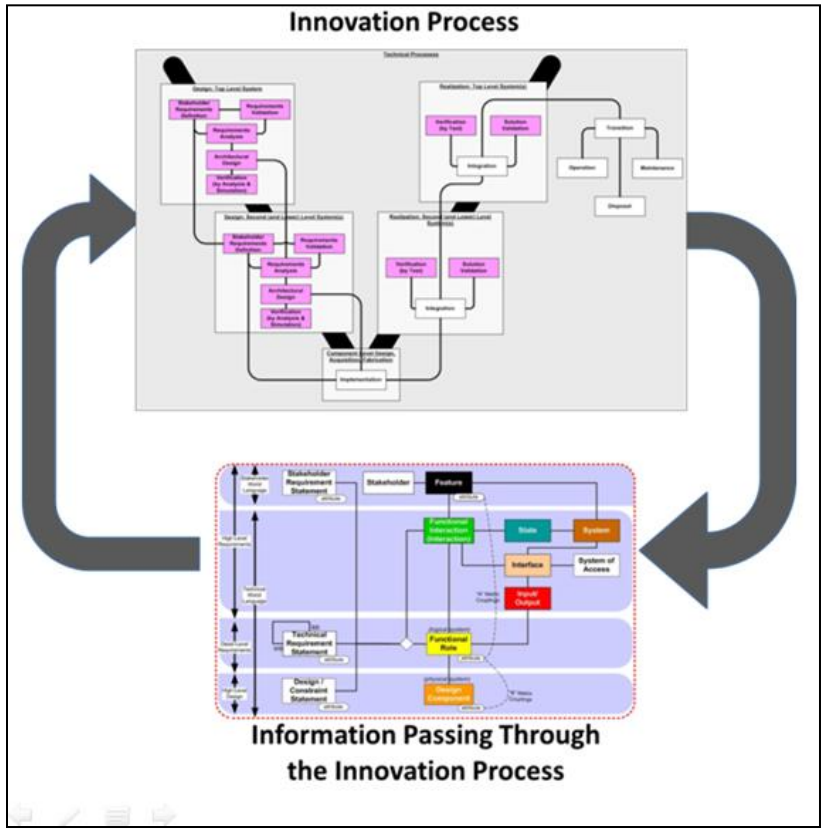
Geometrization of Algebra, by Rene Descartes

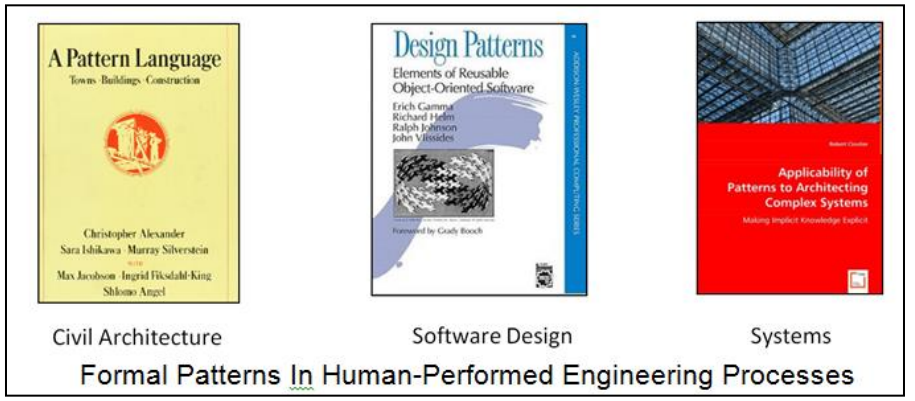


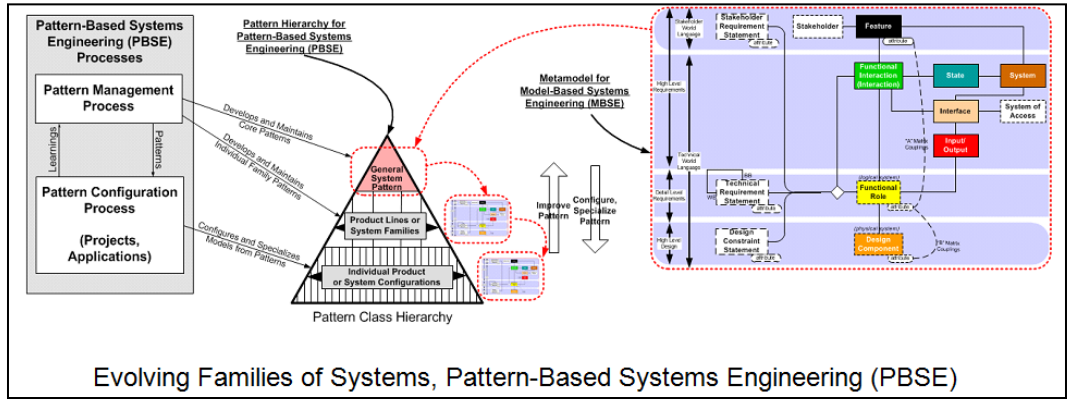
David Hilbert
1862 - 1943

Geometrization of Function Space, by David Hilbert

Maps *vs*. Itineraries -- SE Information *vs*. SE Process

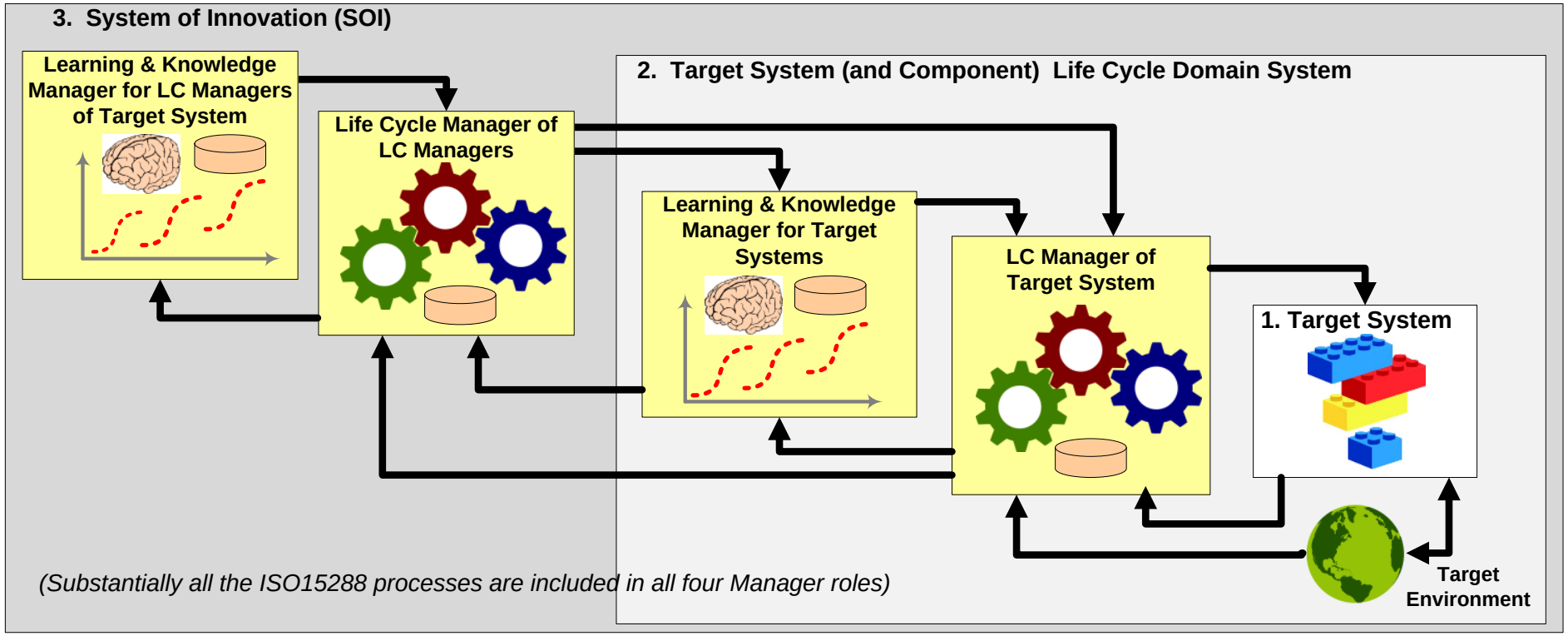






- Model-based Patterns in S*Space.
- Interactions as the basis of all laws of physical sciences.
- Relationships, not procedures, are the fruits of science used by engineers: Newton's laws, Maxwell's Equations.
- Immediate connection to Agility: knowing where you are--starting with better definition of what "where" means. There is a minimal "genome" (S*Metamodel) that provides a practical way to capture, record, and understand—the "smallest model of a system".
- Not giving up process: MBSE/PBSE version of ISO/IEC 15288.

What is the Agile Systems Engineering Life Cycle Pattern?



- A key subset of the ASELCM Pattern is--

  the system reference boundaries . . .

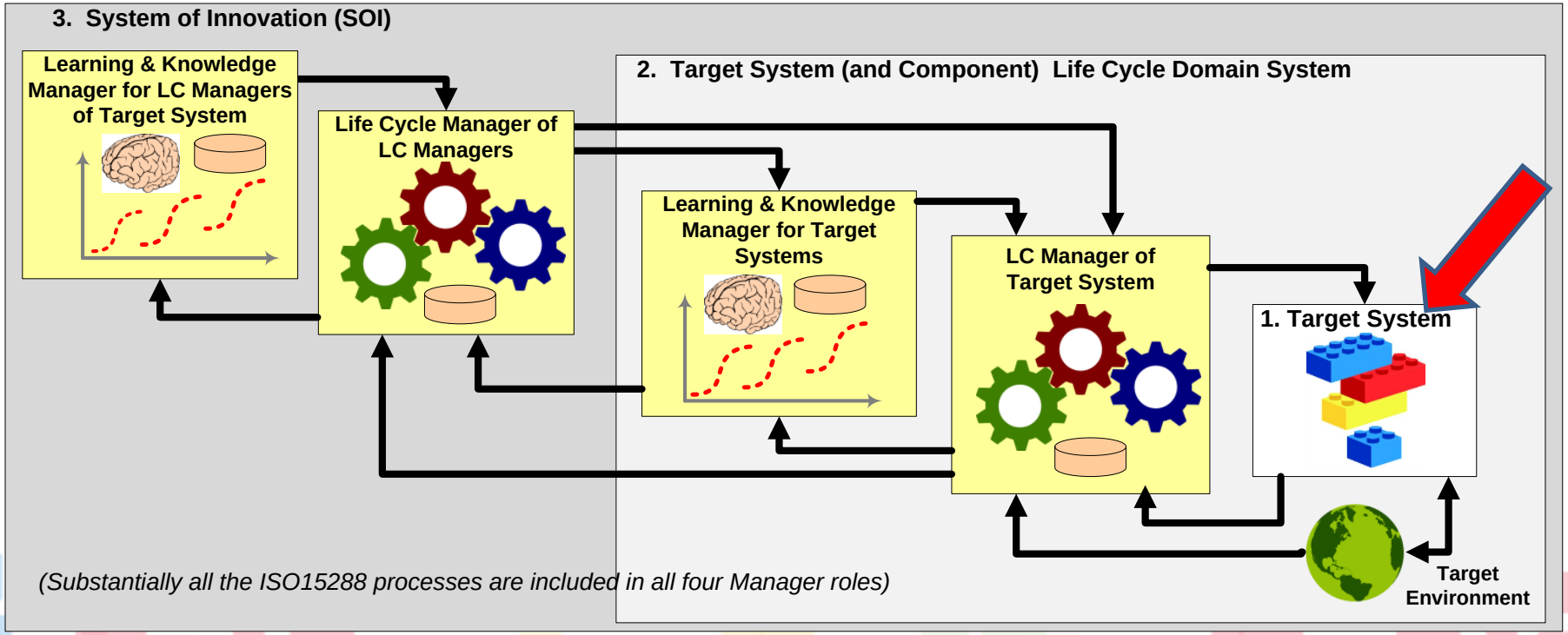
The Agile System Domain Model

- We will particularly refer to **three major system boundaries**:
  - To avoid a confusion bog of loaded terms, we could have just named them “System 1”, “System 2”, and “System 3” and proceeded to define them behaviorally.
  - The definitions are behavioral because these are logical systems, performing defined roles.
  - However, we will also give them more specific names — but make sure you understand the definitions of these systems, which are more important than their names . . .

The Agile System Domain Model

System 1: The **Target System (and Components):** (Definition) The logical system of interest, which results from, or is subject to, innovation.

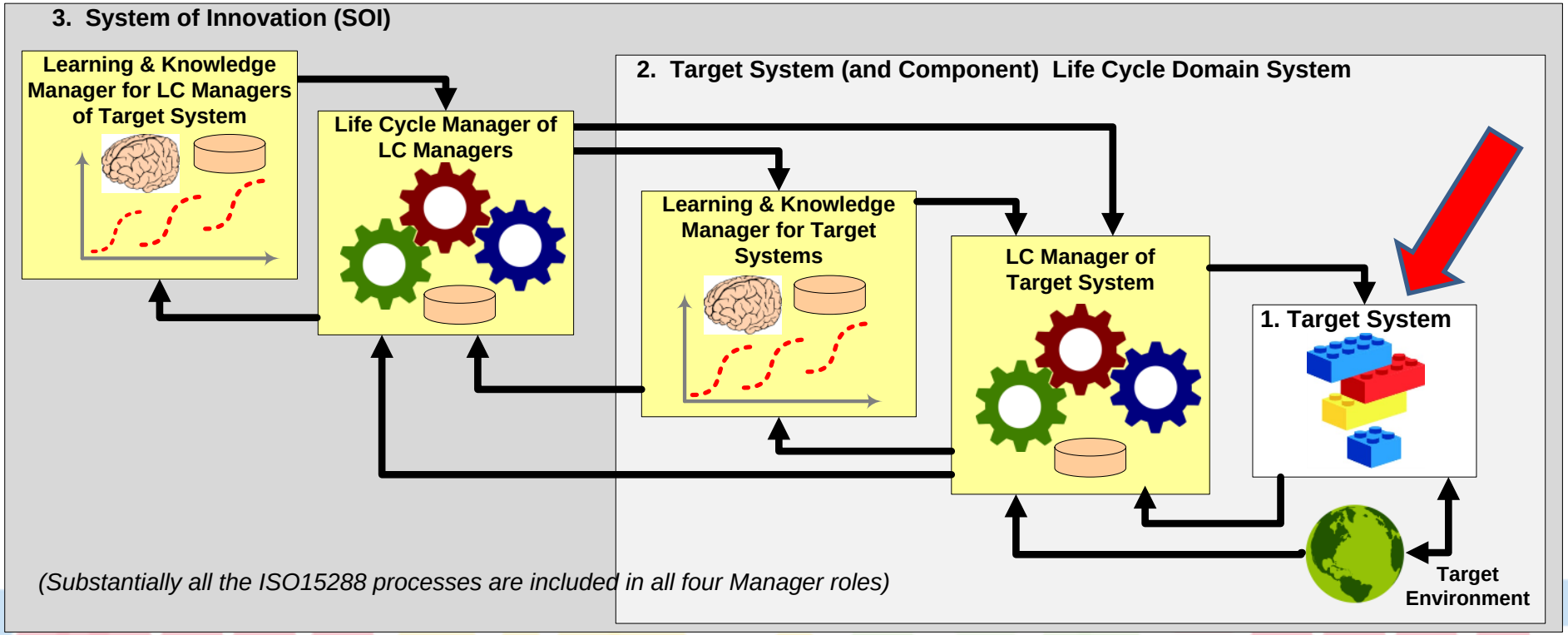
- Its behavior, characteristics, or performance are targets of the innovation (change, adaptation) process we'll introduce later.
- It is potentially agile. (Assertion: for SE to be fully agile, so must its target)
- Examples potentially include aircraft, automobiles, satellites, the human population, software, restaurants.



The Agile System Domain Model

System 1: The **Target System (and Components):** (Definition) The logical system of interest, which results from, or is subject to, innovation.

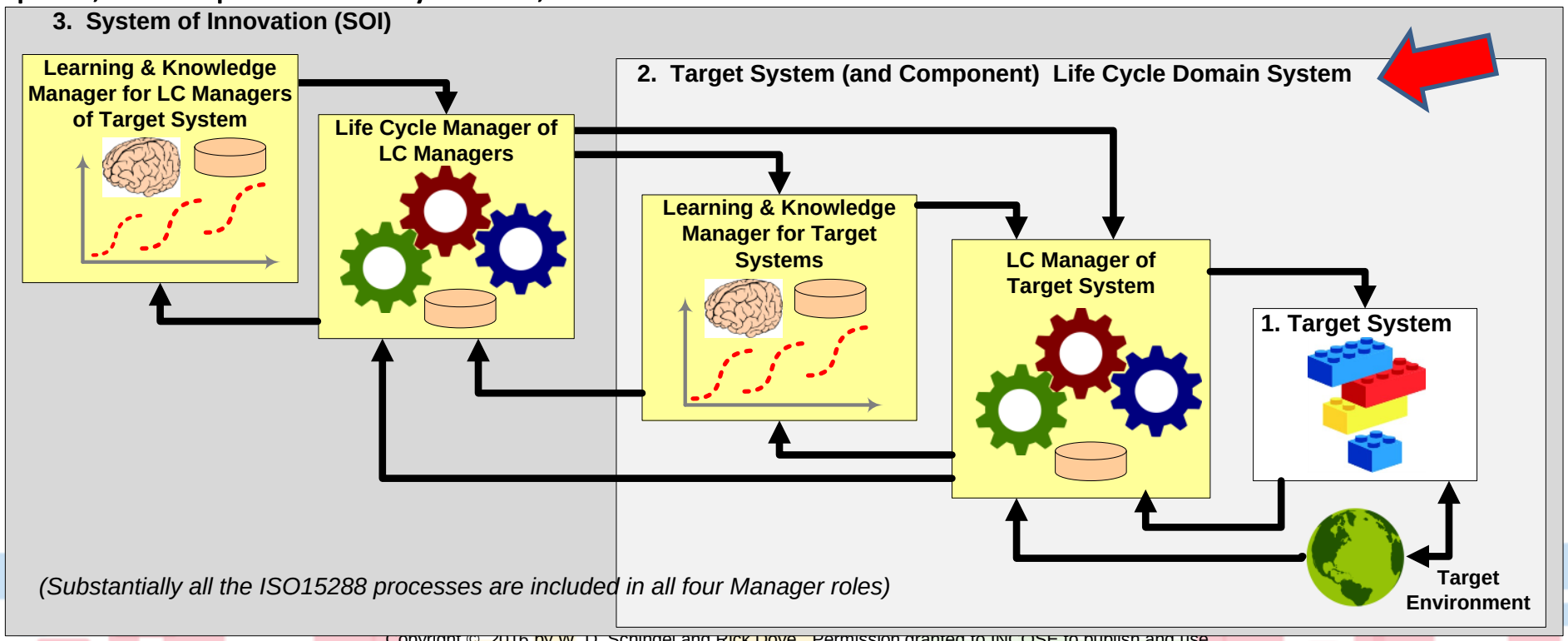
- The Components maintained for integration into a Target System, but not yet integrated, are included in this domain.
- Notice that this idea can apply at multiple additional levels (e.g., System of Systems, System, Subsystem, Component, etc.)



The Agile System Domain Model

System 2: The **Target System (and Component) Life Cycle Domain System:**
(Definition) The logical system within which the Target System will exist during its life cycle, when "in service" or otherwise. This domain includes *all actors with which the Target System will directly interact during its life cycle*:

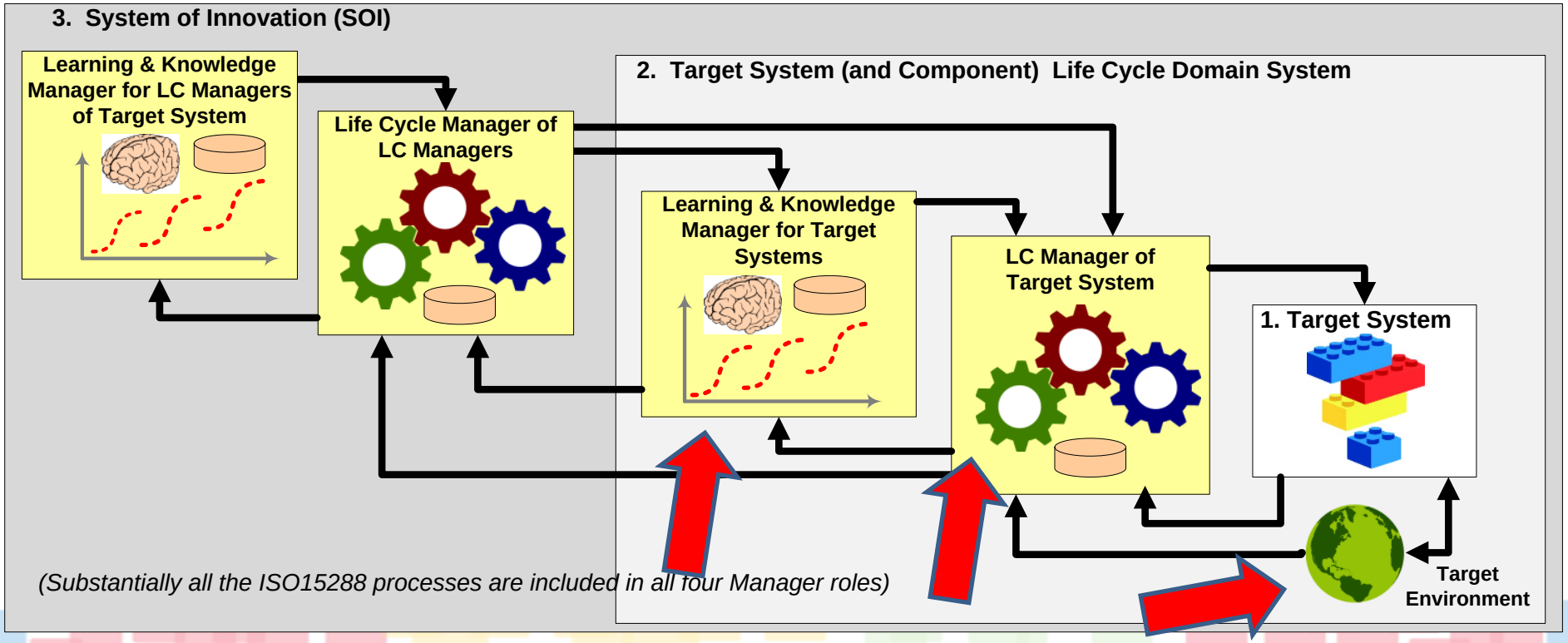
- This includes (among others) any system that directly manages the life cycle of an instance of a Target System (or a Component)—production and integration systems, maintenance, support, and operations systems, and others.



The Agile System Domain Model

The System 2 model recognizes three systems besides the Target System:

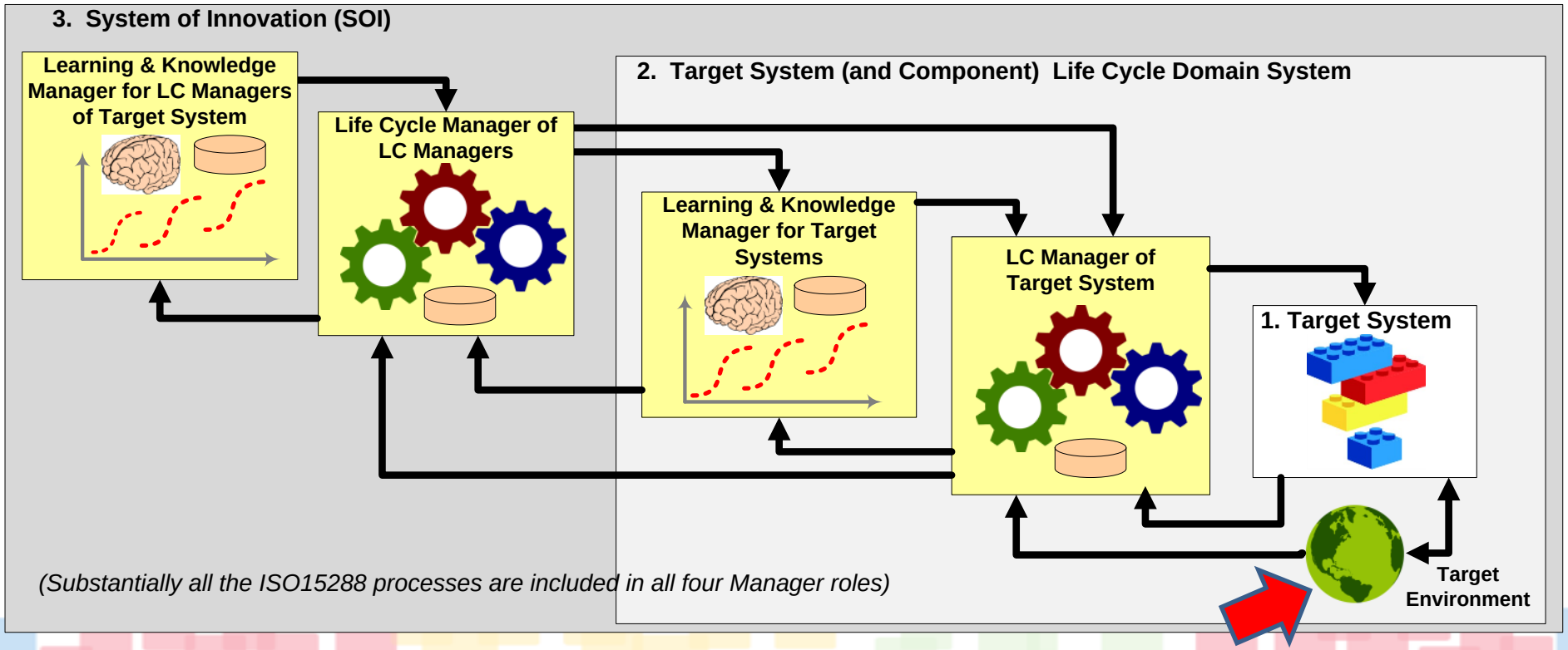
- Target Environment: Target System Life Cycle Domain Actors
- LC Manager of Target System (also manages Target System Components)
- Learning & Knowledge Managers for Target System (and Components)



The Agile System Domain Model

The System 2 model recognizes three systems besides the Target System:

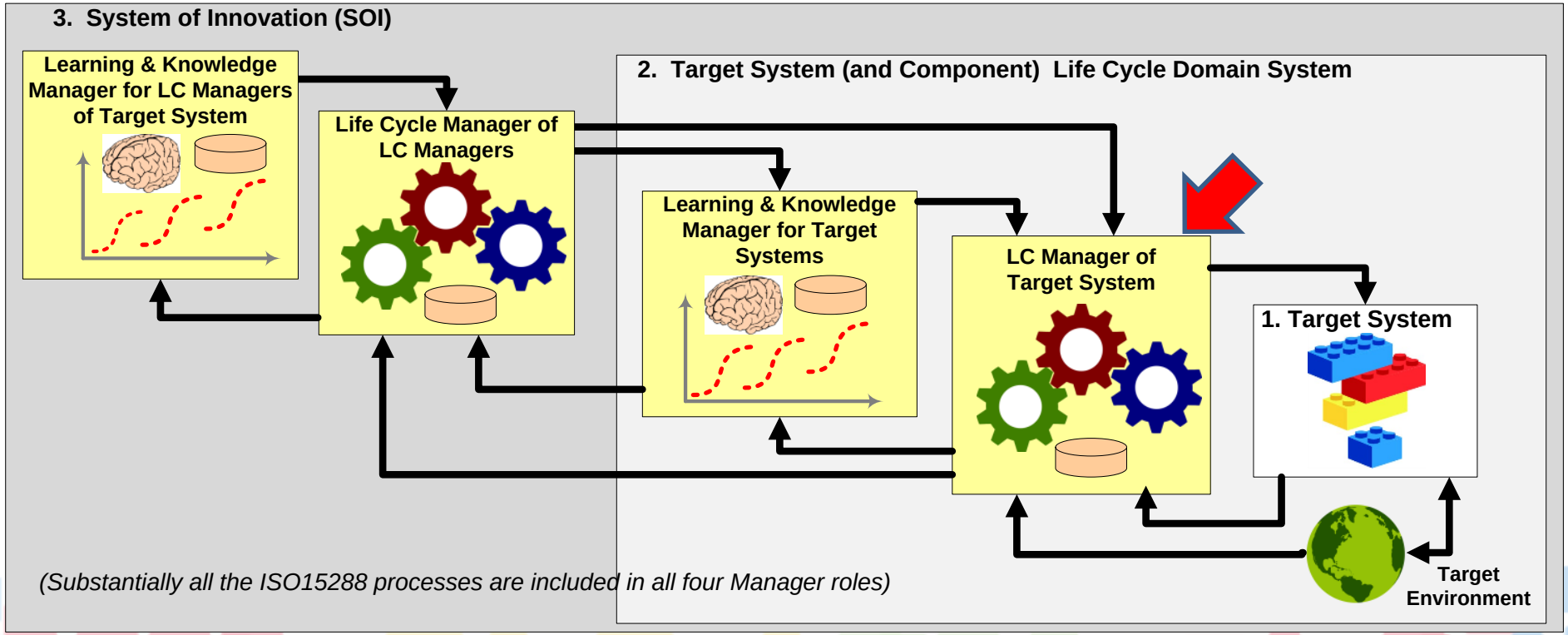
- Target System Life Cycle Domain Actors: All actors with which the Target System will directly interact during its life cycle—those in its operational domain as well as all other direct actors.
- The next system is a special case of those actors . . .



The Agile System Domain Model

The System 2 model recognizes three systems besides the Target System:

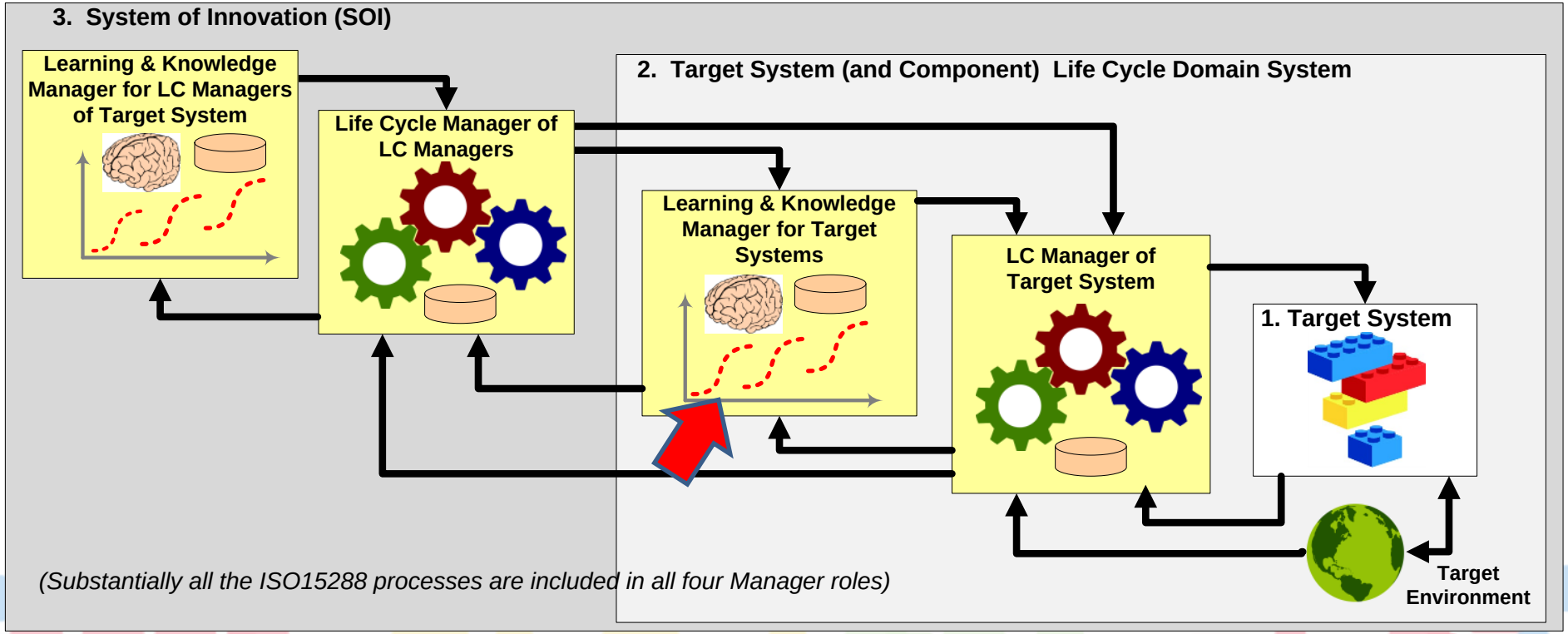
- LC Manager of Target System: Manages all life cycle aspects of the Target System, as recognized by ISO 15288. Note that this is more than just development or systems engineering—it includes manufacturing or acquisition, operations, maintenance, configuration management, and all the ISO System Management Functional Areas.
  - However, it addresses only "already known" aspects of System 1 and Domain Actors—it does not include responsibility of learning new things about them . . .



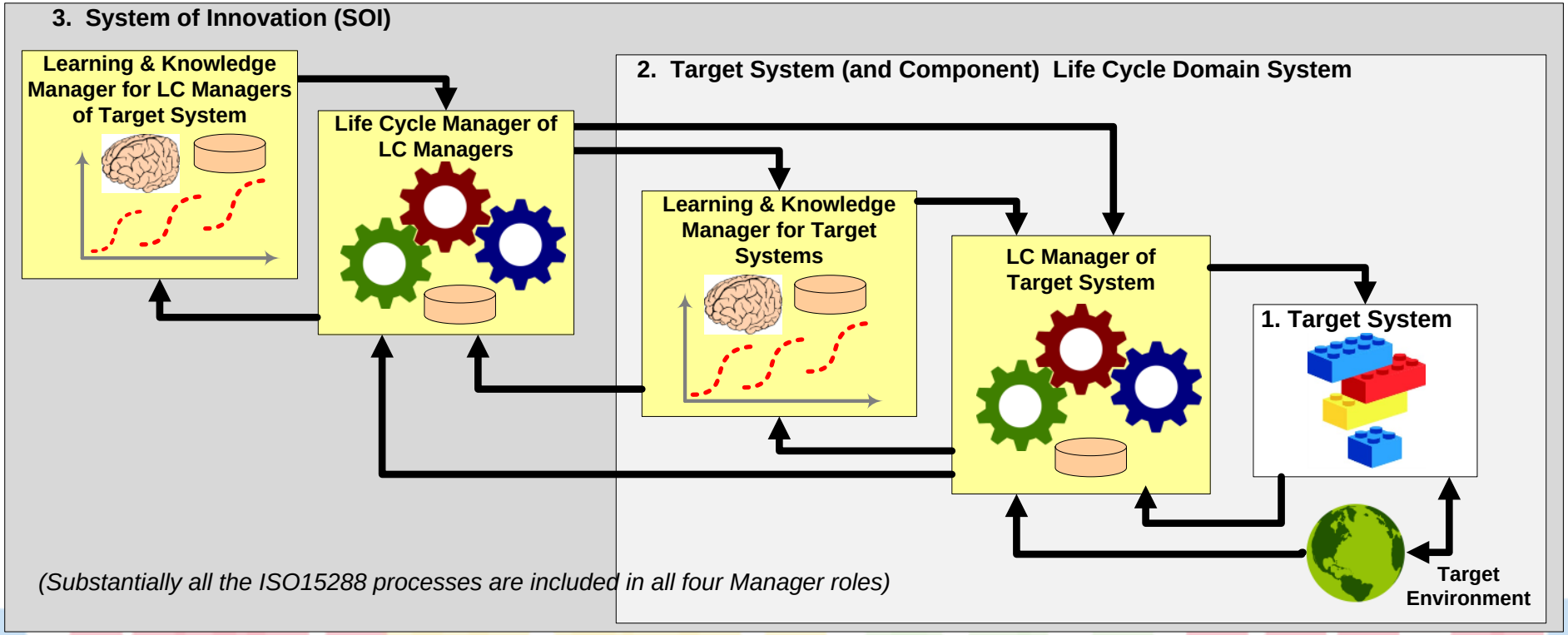
The Agile System Domain Model

The System 2 model recognizes three systems besides the Target System:

- Learning & Knowledge Manager for Target System (and Components): Responsible for learning new things about the Target System, its Components, and its Environment. This may include extraction of patterns or other knowledge from observations, planning experiments and extracting conclusions from their results, and other forms of learning. It also includes responsibility for accumulation and persistent memory of those learnings, and for providing the resulting knowledge for use by the LC Managers of the Target System.

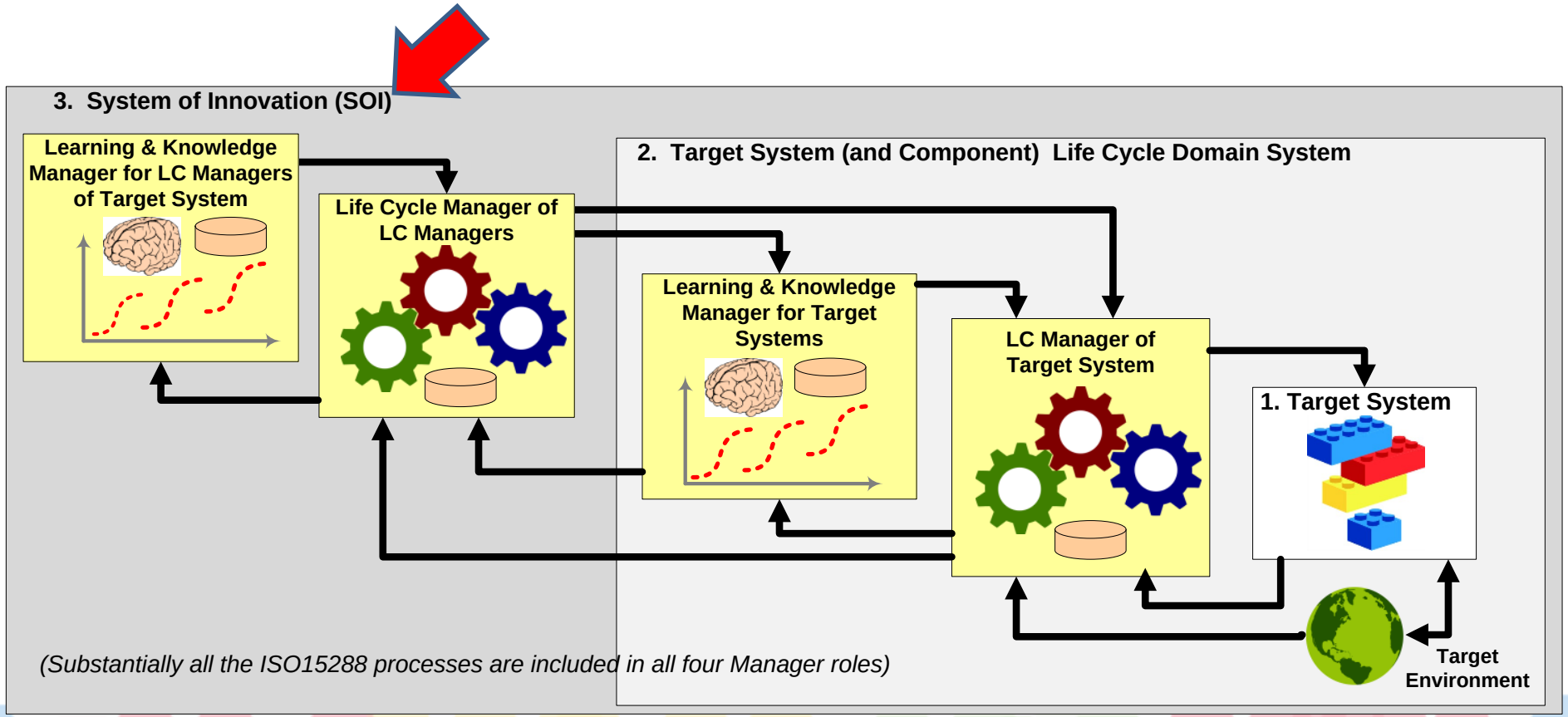


- Again, remember that these are logical (behavioral) roles. In realized systems, a single physical system may behave as both a Target System and a system that produces, modifies, reconfigures, or otherwise manages a Target System, by having roles from each allocated to it.
- For purposes of this logical roles description, they have been identified separately.
- We introduce the physical components into the model later.



System 3: The **System of Innovation:** (Definition) The logical system responsible for managing the life cycles of instances of any (System 2) Target System LC Manager.

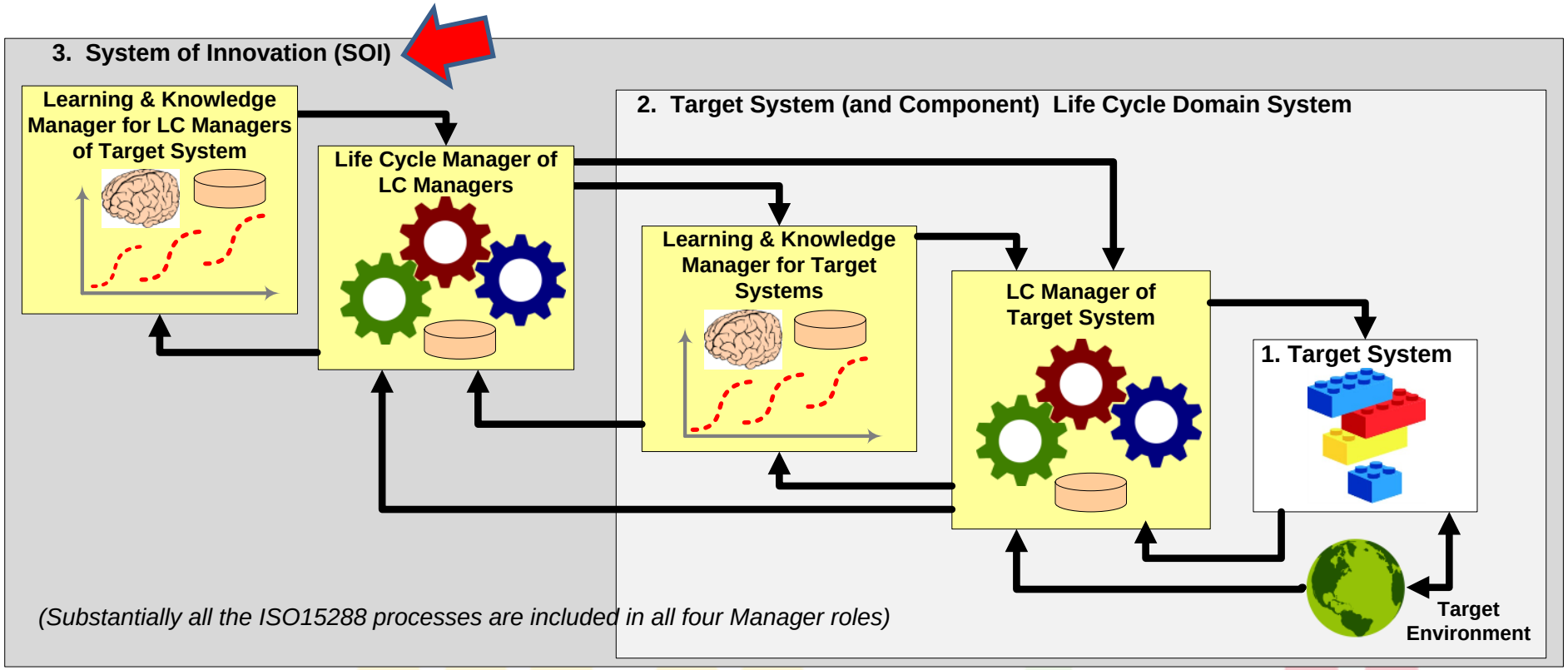
- (Recall that those System 2 Target System LC Managers include Target System development, production, integration, maintenance, operations, and other management systems.)



The Agile System Domain Model

The System 3 model recognizes two sub-systems of System 3:

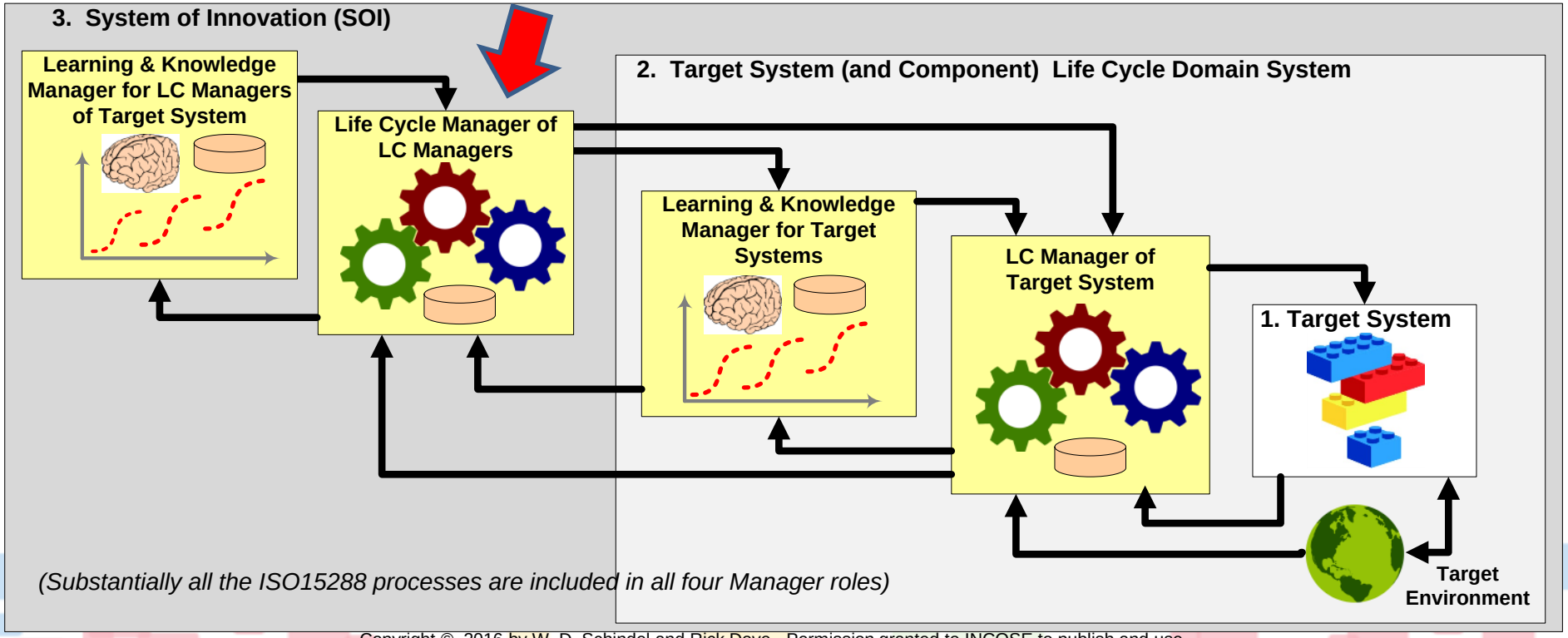
- Life Cycle Manager of LC Managers
- Learning & Knowledge Managers for LC Managers of Target Systems



The Agile System Domain Model

The System 3 model recognizes two sub-systems of System 3:

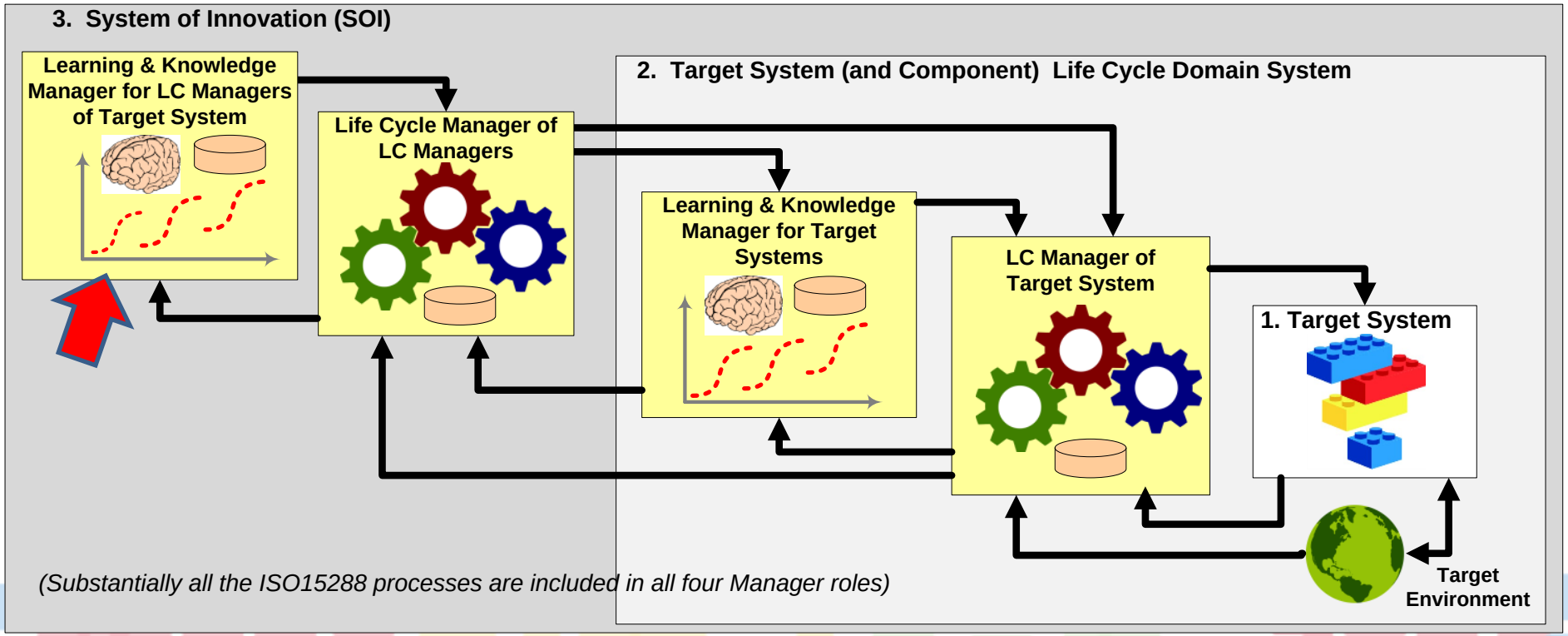
- Life Cycle Manager of LC Managers: Manages all life cycle aspects of the LC Managers of Target Systems, as recognized by ISO 15288. Note that this is more than just development or systems engineering—it includes their design or acquisition, maintenance, configuration management, and all the ISO System Management Functional Areas.
  - However, it addresses only "already known" aspects of the LC Managers in System 2—it does not include responsibility of learning new things about them . . .



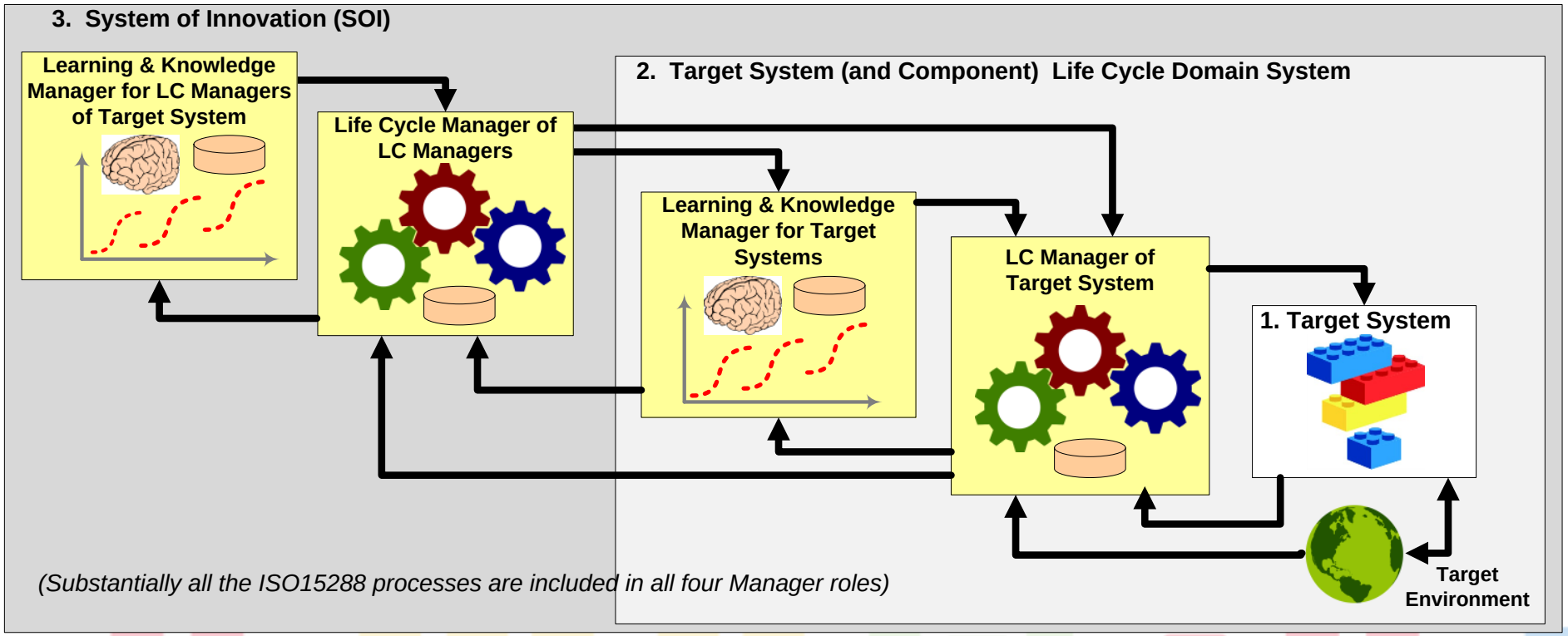
The Agile System Domain Model

The System 3 model recognizes two sub-systems of System 3:

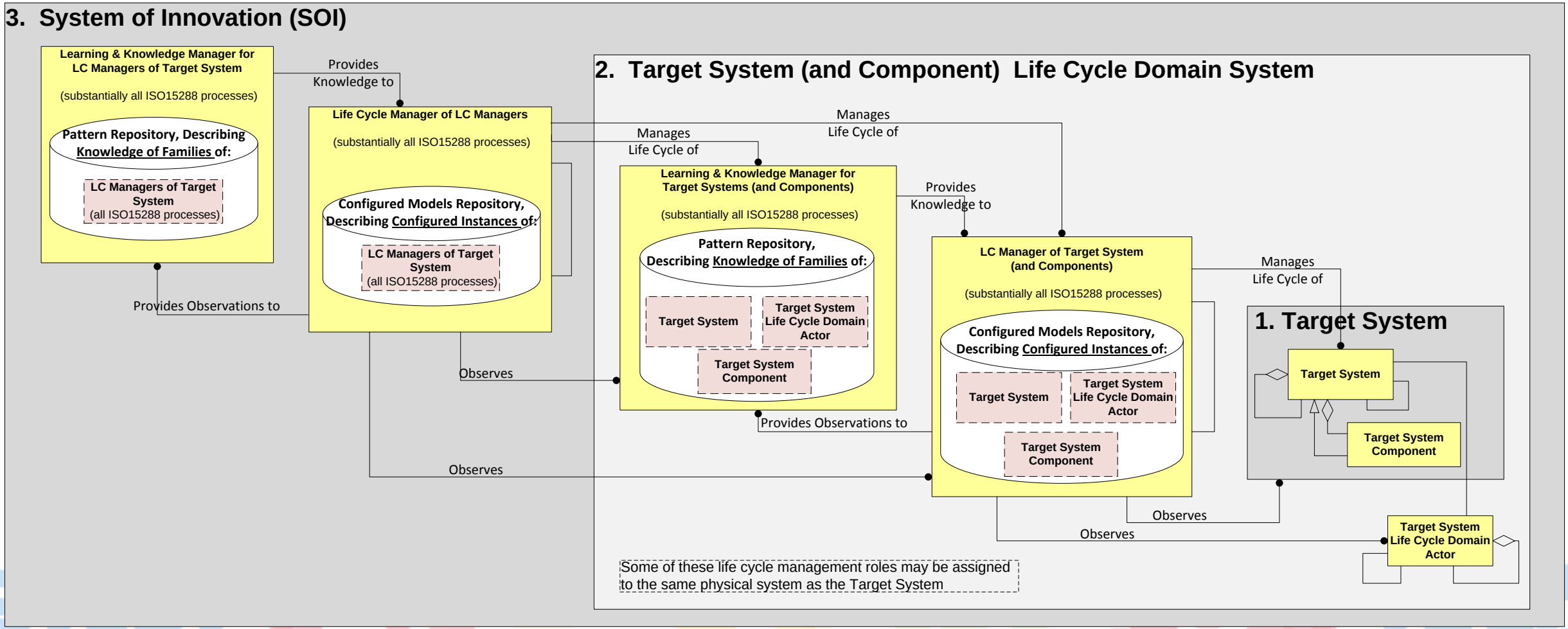
- Learning & Knowledge Managers for LC Managers of Target Systems: Responsible for learning new things about the LC Managers in System 2. This may include extraction of patterns or other knowledge from observations, planning experiments and extracting conclusions from their results, and other forms of learning. It also includes responsibility for accumulation and persistent memory of those learnings, and for providing the resulting knowledge for use by the Life Cycle Manager of the LC Managers.



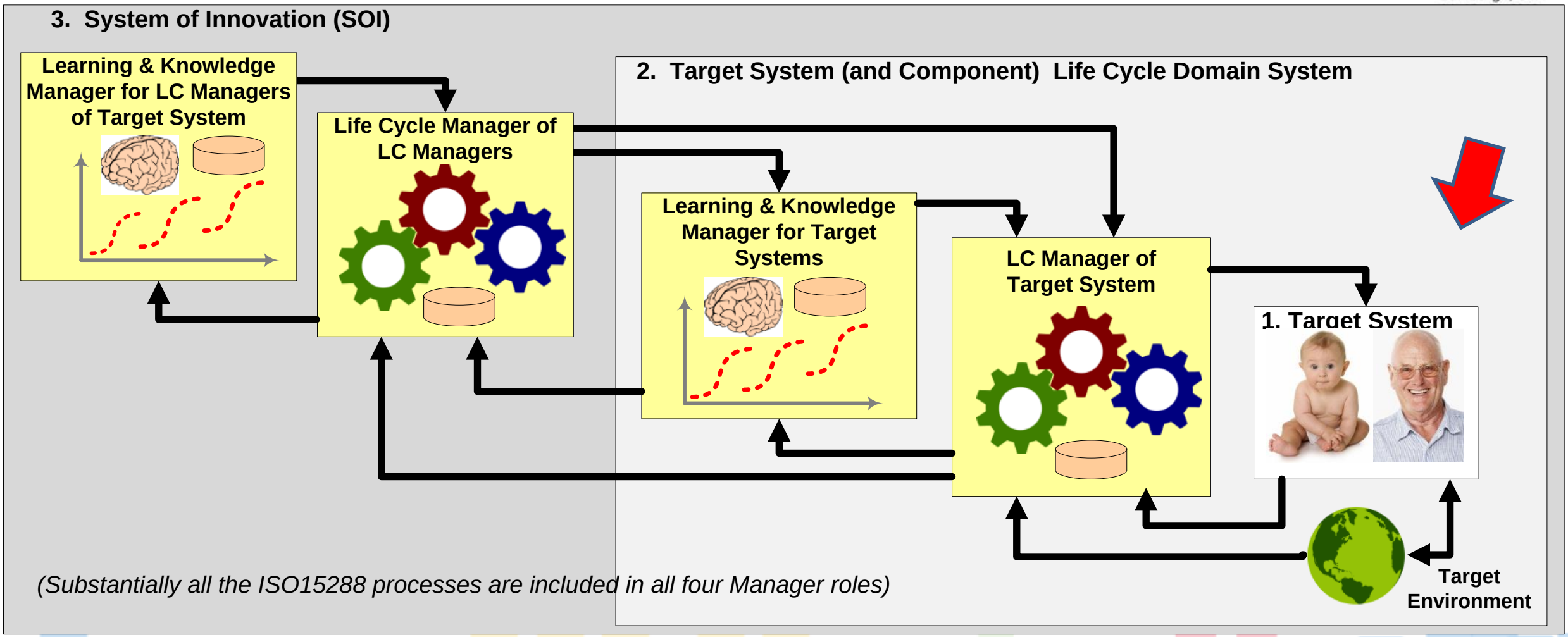
- Summary so far:
  - System 2, the Target System Life Cycle Domain System produces and modifies instances of System 1, the Target Systems (and Components), and also learns new things about System 1 and its environment.
  - System 3, the System of Innovation, produces and modifies instances of System 2, the Target LC Managers, and also learns new System 2 things.



Behind the “iconic” diagram, there is a formal MBSE model that describes the ASELCM Pattern

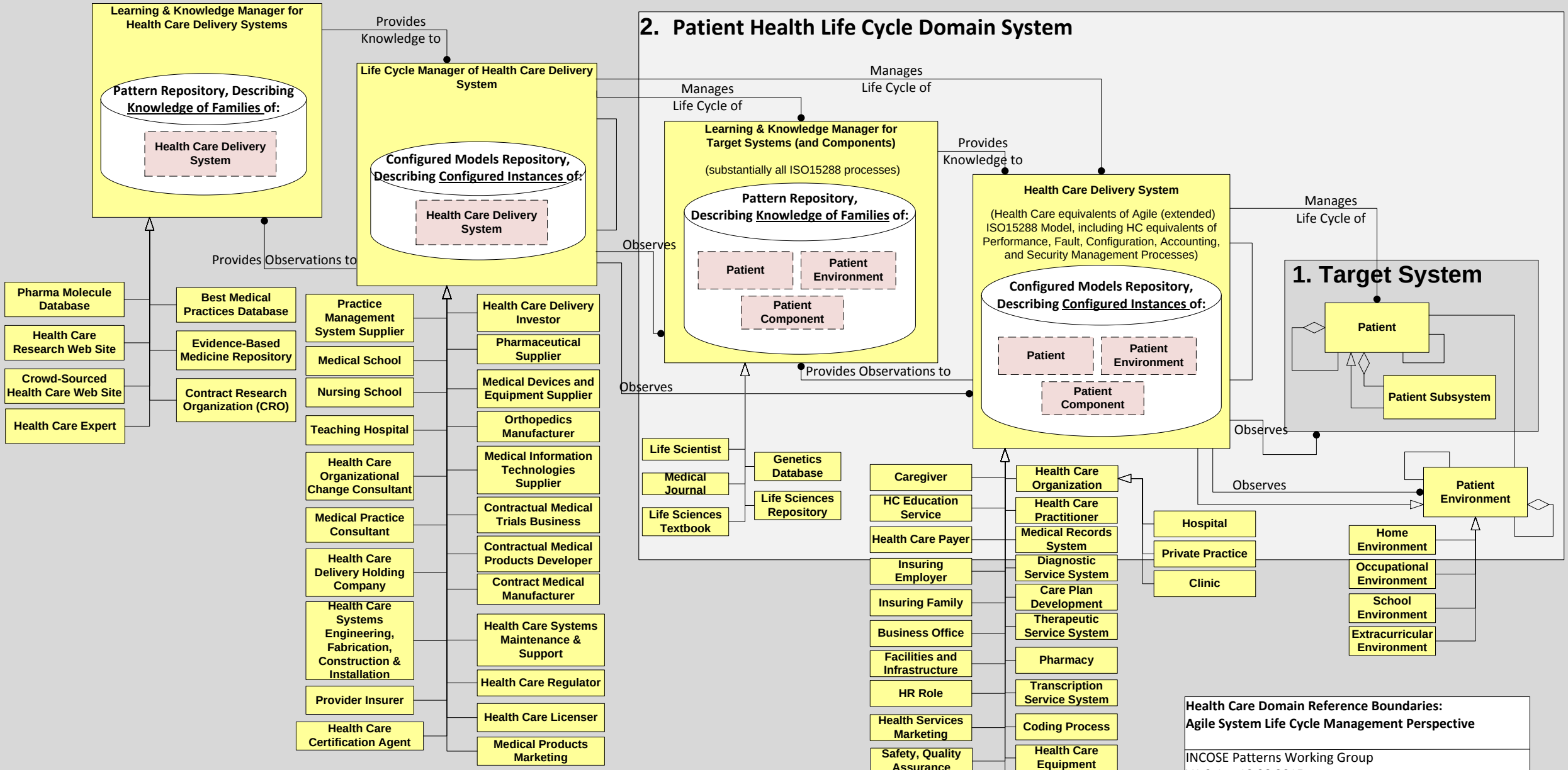


Example: Applying the ASELCM Pattern to Plan Agility Improvement in Health Care Systems



(From INCOSE 2016 Agile Health Care Systems Conference)

3. Health Care System of Innovation (SOI)



Health Care Domain Reference Boundaries:
Agile System Life Cycle Management Perspective

INCOSE Patterns Working Group
V1.3.1 10.28.2015

Example: Health care domain, top level

A Breakout Exercise at the INCOSE 2016 Agile Health Care Systems Conference:

Assessing and Planning Agility Improvements in Health Care Systems

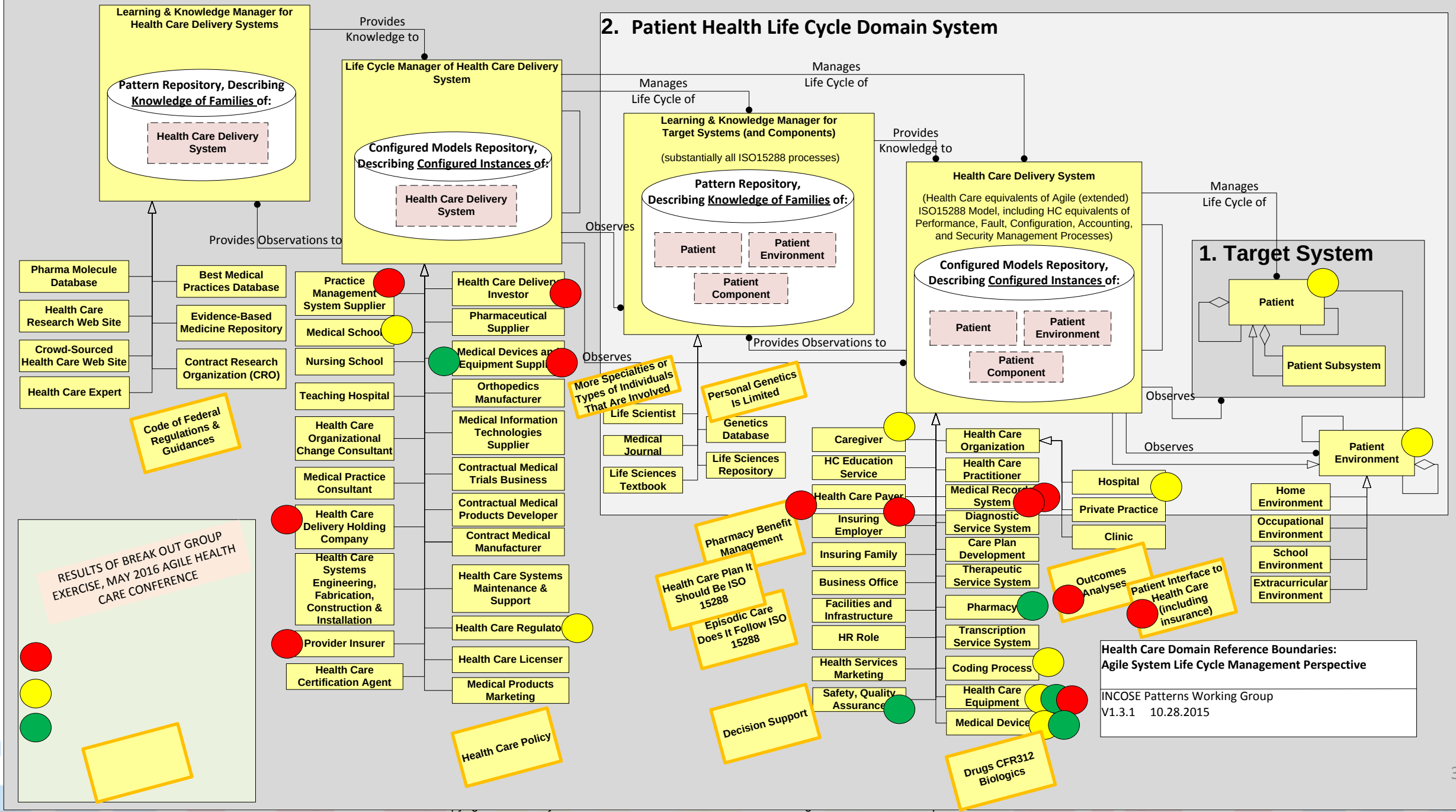
- Directions:
  - Break into teams and discuss the following, then . . .
  - In the domain model, identify the 5 highest cases of:

Sticky Dots

- Needs for improved future agility (even if most difficult)
- Opportunities for improved future agility (low-hanging fruit)
- Already accomplished examples of improved agility progress (e.g., defense theater medicine, device software, etc.)

Sticky note In the domain model, identify potential corrections or improvements to the model

3. Health Care System of Innovation (SOI)



Concluding Discussion, Q&A

- The ASELCM Pattern also contains a lot more detail levels
- The ASELCM Pattern is a work in process
- Being validated and used in the ASELCM Project
- You are invited to participate!
-
-
-
-

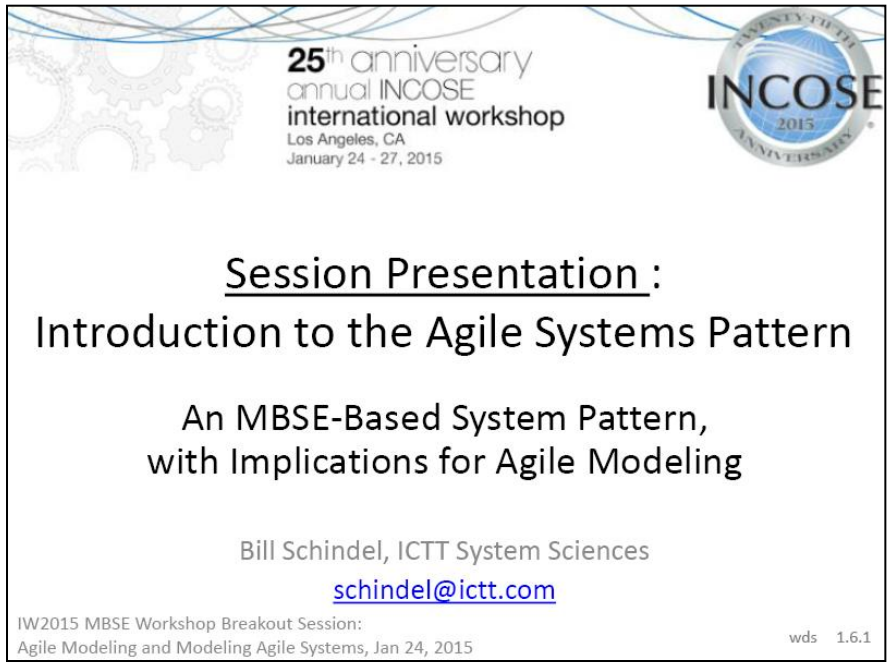
Primary References

1. Rick Dove, Ralph LaBarge, “Fundamentals of Agile Systems Engineering—Part 1” and “Part 2”, INCOSE IS2014, July, 2014.
2. Rick Dove, “Agile Systems 101”, “102” and “103”, Paradigm Shift, International, http://www.parshift.com/s/AgileSystems-101.pdf
3. ----------, *Response Ability: The Language, Structure, and Culture of the Agile Enterprise,* Wiley, 2001.
4. Martin Fowler and Jim Highsmith. “The Agile Manifesto”. *Dr. Dobb's Journal*, August, 2001. www.drdobbs.com/open-source/the-agile-manifesto/184414755 .
5. J. Highsmith. *Agile Software Development Ecosystems*. Addison-Wesley Professional, 2002.
6. W. Schindel, “What Is the Smallest Model of a System?”, *Proc. of the INCOSE 2011 International Symposium*, International Council on Systems Engineering (2011).
7. Bill Schindel, Troy Peterson, “Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques”, in Proc. of INCOSE 2013 International Symposium, Tutorial, June, 2013.
8. W. Schindel, “Maps or Itineraries?: A Systems Engineering Insight from Ancient Navigators”, to appear in Proc. of INCOSE International Symposium 2015, July, 2015.
9. ----------, “System Life Cycle Trajectories: Tracking Innovation Paths Using System DNA”, to appear in Proc. of INCOSE International Symposium 2015, July, 2015.
10. Schindel and Beihoff: “Systems of Innovation I: Models of Their Health and Pathologies”, Proc. of INCOSE International Symposium, 2012.
11. W. Schindel, “Systems of Innovation II: The Emergence of Purpose”, Proceedings of INCOSE 2013 International Symposium (2013)

Secondary References

1. Rick Dove, “Agile Systems and Processes—Driving Architecture with ConOps and Response Situation Analysis (Agile 102)”, Paradigm Shift, International, September 16, 2013.
2. Rick Dove, “Security R Us: Systems Engineering is the High Ground”, INCOSE Biomedical Healthcare Working Group, April 24, 2014.
3. W. Schindel, “Requirements statements are transfer functions: An insight from model-based systems engineering”, *Proceedings of INCOSE 2005 International Symposium*, (2005).
4. “OMG Systems Modeling Language, Version 1.3”, Object Management Group, June, 2012.
5. W. Schindel, and V. Smith, “Results of applying a families-of-systems approach to systems engineering of product line families”, SAE International, Technical Report 2002-01-3086 (2002).
6. W. Schindel, “Pattern-Based Systems Engineering: An Extension of Model-Based SE”, INCOSE IS2005 Tutorial TIES 4, 2005.
7. J. Bradley, M. Hughes, and W. Schindel, “Optimizing Delivery of Global Pharmaceutical Packaging Solutions, Using Systems Engineering Patterns” Proceedings of the INCOSE 2010 International Symposium (2010).
8. W. Schindel, “The Impact of ‘Dark Patterns’ On Uncertainty: Enhancing Adaptability In The Systems World”, in Proc. of INCOSE Great Lakes 2011 Regional Conference on Systems Engineering, Dearborn, MI, 2011
9. INCOSE/OMG Patterns Working Group 2013-14 http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
10. W. Schindel, S. Peffers, J. Hanson, J. Ahmed, W. Kline, “All Innovation is Innovation of Systems : An Integrated 3-D Model of Innovation Competencies ”, Proc. of ASEE 2011 Conference, American Association for Engineering Education, (2011).
11. ISO/IEC 15288: Systems Engineering—System Life Cycle Processes. International Standards Organization (2014).
12. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Version 4, International Council on Systems Engineering (2014).
13. “A World in Motion: Systems Engineering Vision 2025”, INCOSE, 2014.
14. “Instrument Unit Fact Sheet: Saturn V News Reference”, IBM Federal Systems Division, 1968.
15. W. David Woods, Kenneth D. MacTaggart and Frank O'Brien, “Apollo 11 Flight Journal: Day 2, Part 1: Mid Course Correction”, http://history.nasa.gov/ap11fj/05day2-mcc.htm , updated 2009.

Back Up

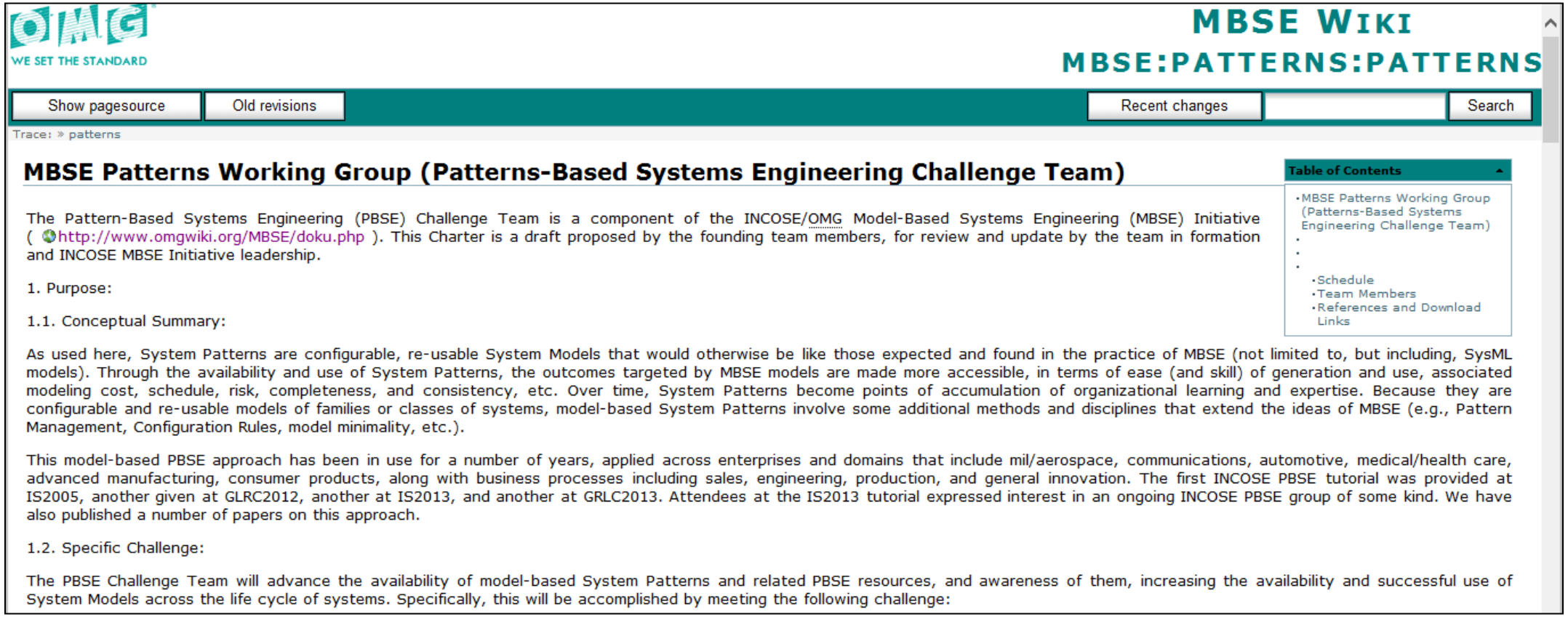


See --

http://www.omgwiki.org/MBSE/doku.php?id=mbse:incose_mbse_iw_2015:breakout_out_session_agile_modeling

What is the INCOSE MBSE Patterns Working Group?

http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns



OMG
WE SET THE STANDARD

MBSE WIKI
MBSE:PATTERNS:PATTERNS

Show pagesource | Old revisions | Recent changes | Search

Trace: » patterns

MBSE Patterns Working Group (Patterns-Based Systems Engineering Challenge Team)

Table of Contents
- MBSE Patterns Working Group (Patterns-Based Systems Engineering Challenge Team)
- Schedule
- Team Members
- References and Download Links

The Pattern-Based Systems Engineering (PBSE) Challenge Team is a component of the INCOSE/OMG Model-Based Systems Engineering (MBSE) Initiative (http://www.omgwiki.org/MBSE/doku.php). This Charter is a draft proposed by the founding team members, for review and update by the team in formation and INCOSE MBSE Initiative leadership.

1. Purpose:

1.1. Conceptual Summary:

As used here, System Patterns are configurable, re-usable System Models that would otherwise be like those expected and found in the practice of MBSE (not limited to, but including, SysML models). Through the availability and use of System Patterns, the outcomes targeted by MBSE models are made more accessible, in terms of ease (and skill) of generation and use, associated modeling cost, schedule, risk, completeness, and consistency, etc. Over time, System Patterns become points of accumulation of organizational learning and expertise. Because they are configurable and re-usable models of families or classes of systems, model-based System Patterns involve some additional methods and disciplines that extend the ideas of MBSE (e.g., Pattern Management, Configuration Rules, model minimality, etc.).

This model-based PBSE approach has been in use for a number of years, applied across enterprises and domains that include mil/aerospace, communications, automotive, medical/health care, advanced manufacturing, consumer products, along with business processes including sales, engineering, production, and general innovation. The first INCOSE PBSE tutorial was provided at IS2005, another given at GLRC2012, another at IS2013, and another at GRLC2013. Attendees at the IS2013 tutorial expressed interest in an ongoing INCOSE PBSE group of some kind. We have also published a number of papers on this approach.

1.2. Specific Challenge:

The PBSE Challenge Team will advance the availability of model-based System Patterns and related PBSE resources, and awareness of them, increasing the availability and successful use of System Models across the life cycle of systems. Specifically, this will be accomplished by meeting the following challenge:

MBSE Patterns Working Group's PBSE Methodology Summary for INCOSE includes overview and many references:

"Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models", INCOSE Patterns Challenge Team, 2015.

Introduction to INCOSE MBSE Patterns Working Group

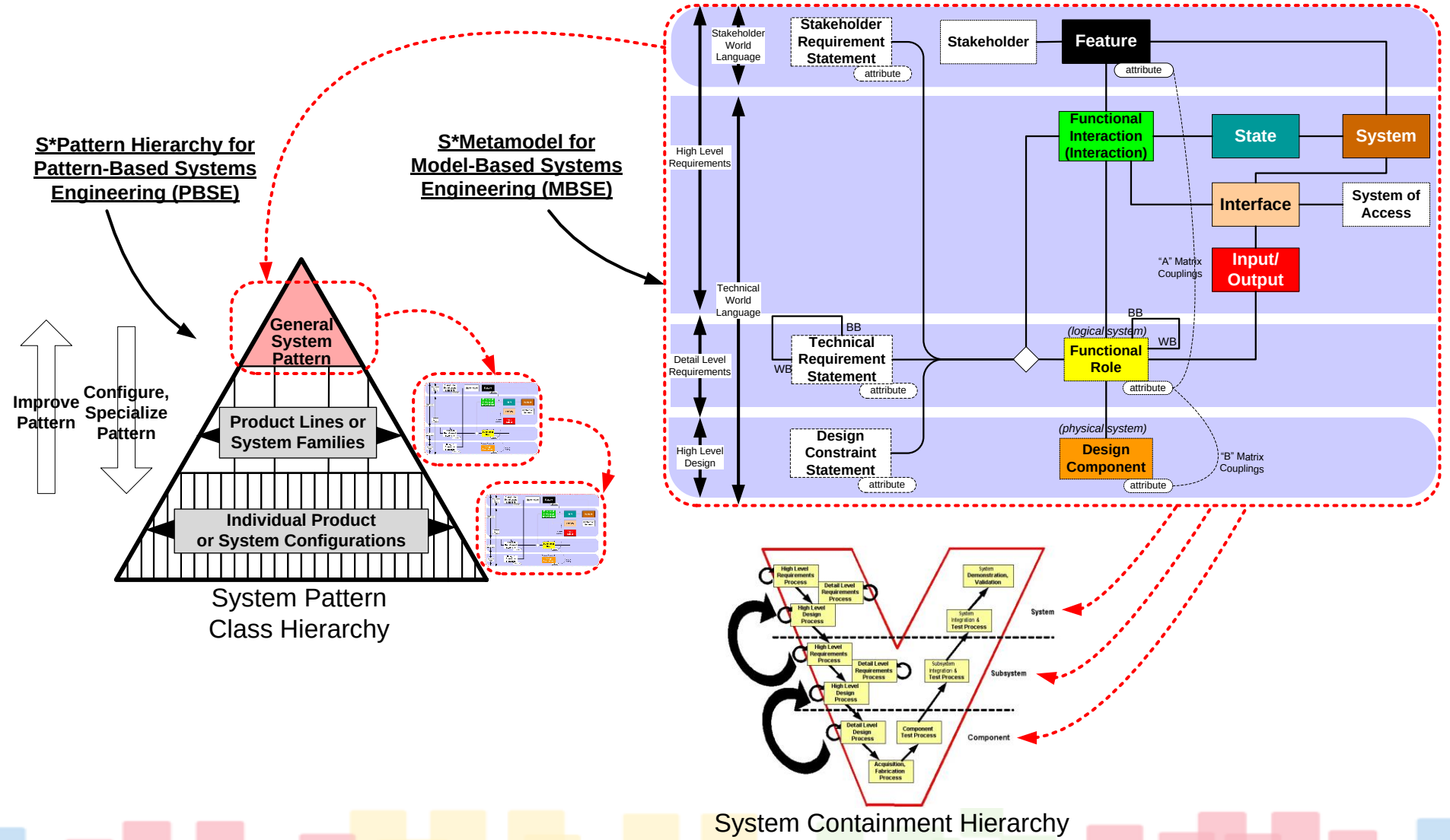
- Started in 2013, meeting several times a year, membership across domains.
- Team Co-chairs: Bill Schindel, Troy Peterson
- Eleven accepted IS2015+IS2016 Patterns Team member papers.
- Re-usable, configurable, MBSE models (“Patterns”).
- Based on S*Metamodel.
- Language and tool independent—frequently in SysML.
- Methodology practiced across domains ~ 20 years.
- For more information . . .

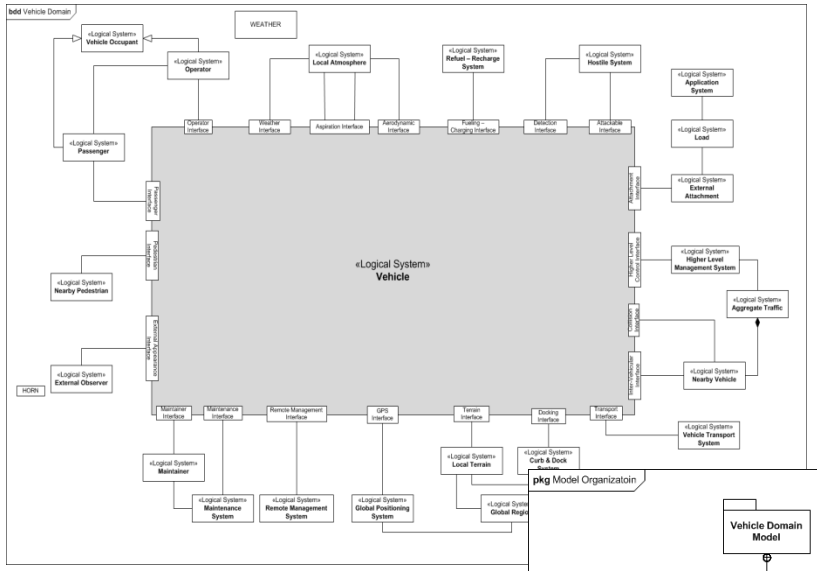
http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

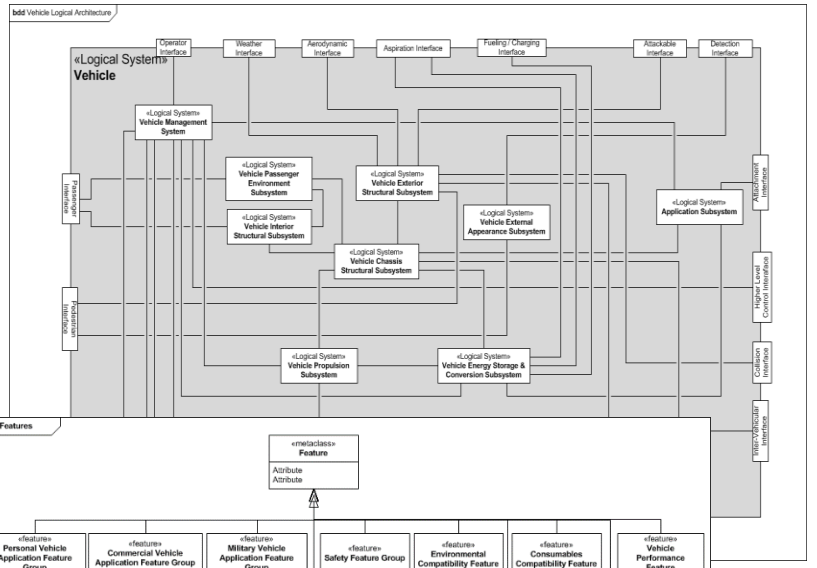
Cooperative cross-team/working group projects

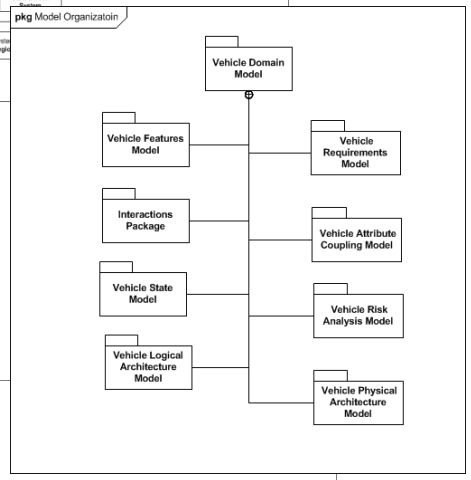
- The Patterns Working Group has been reaching out to other INCOSE and industry working groups:
  - Joint projects of interest.
- Examples:
  - SoS Working Group
  - Agile Systems Working Group
  - PLE Working Group
  - Others in discussion for 2016-2017

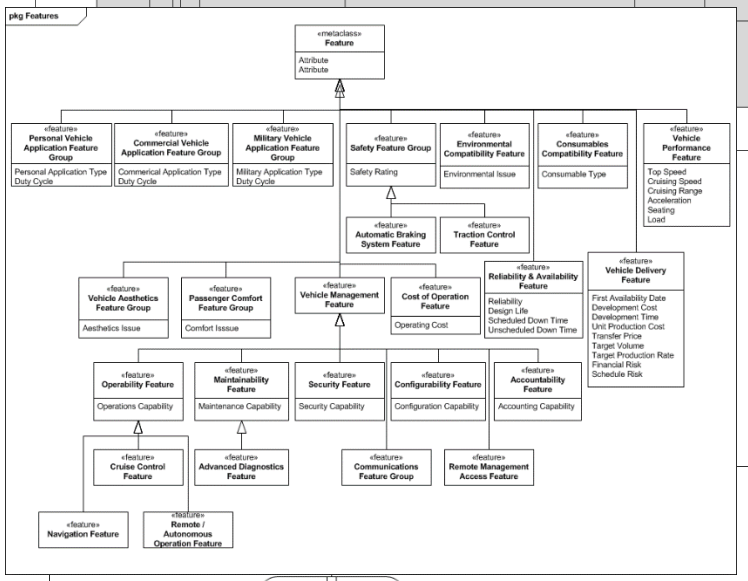
S*Models, S*Patterns

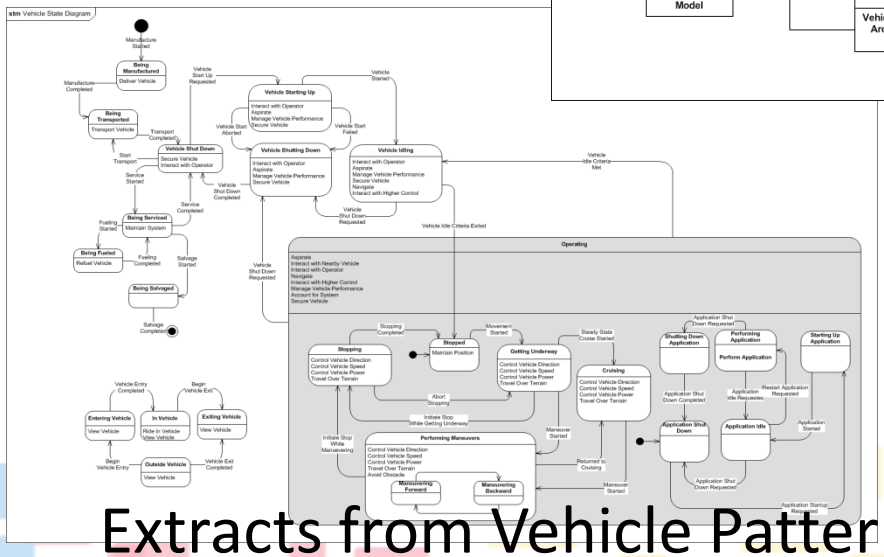


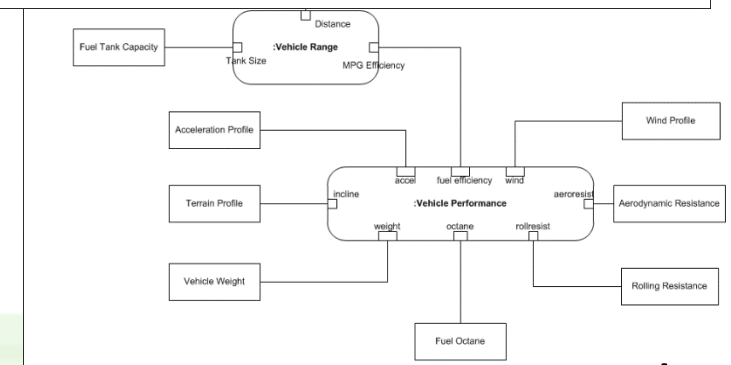










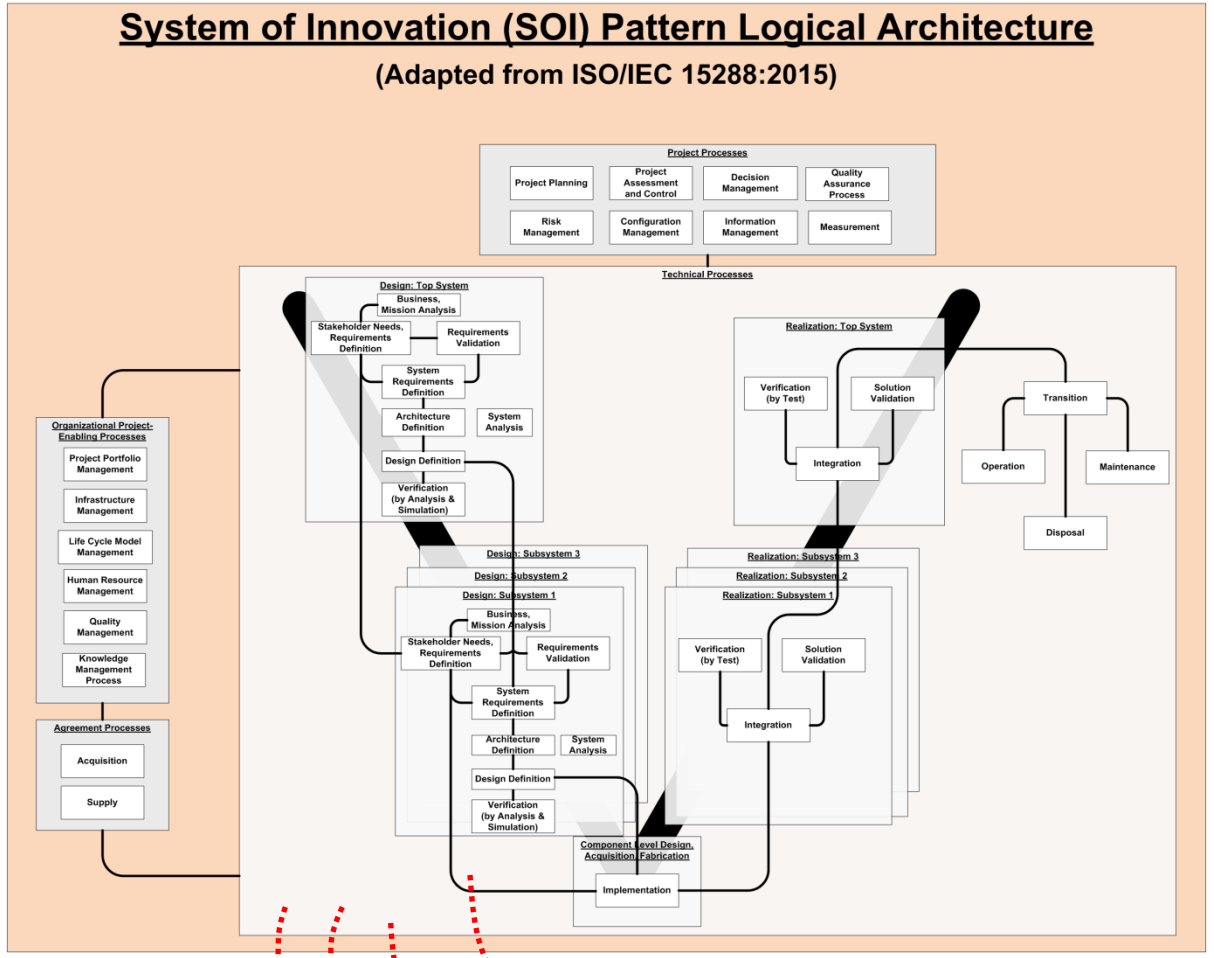


Extracts from Vehicle Pattern in 2013 INCOSE PBSE Tutorial

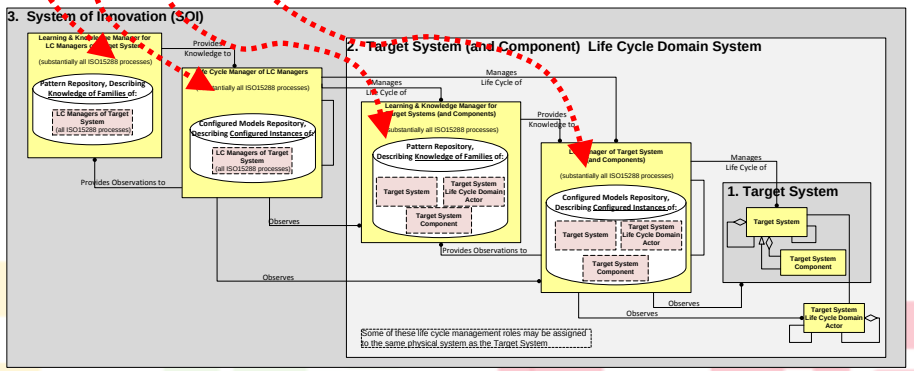
Relating Scrum and ISO 15288 Process Models

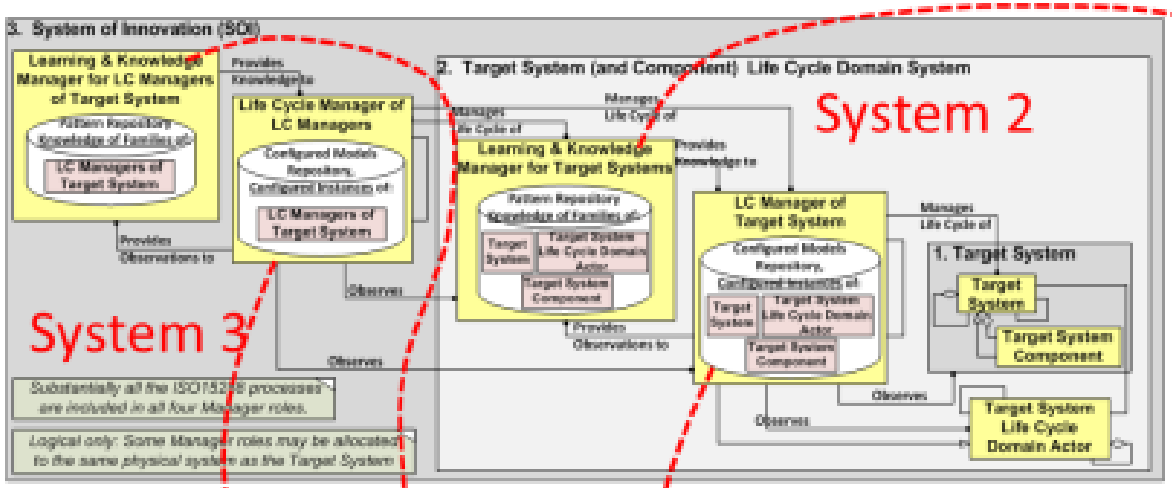
- More Than One Representation (Model View) of the Same Underlying (Process) Reality . . .

ISO15288
Reference Processes

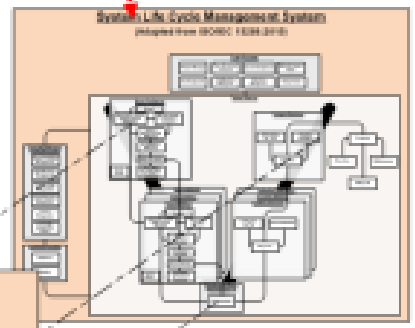


ISO15288 Technical Processes appear in System 2 (for target) and System 3 (for LC managers), as (potentially concurrent) “Vee” processes, and learning sources.



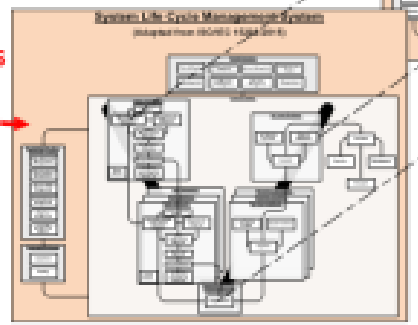


Performs most of these ISO processes, to manage what is being learned about S1 space.

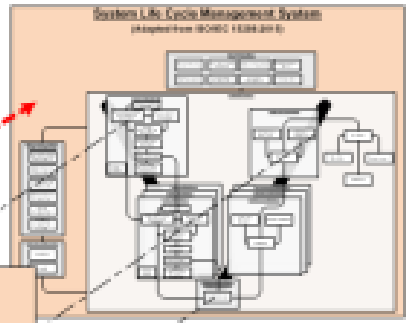


Manages Target System Family Patterns

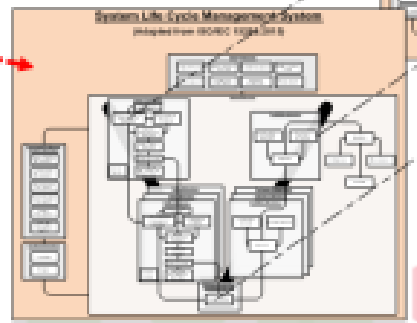
Performs most of these ISO processes, to manage S1 instances , using S1 patterns



Manages LC Management System Family Patterns



Performs most of these ISO processes, to manage S2 instances , using S2 patterns



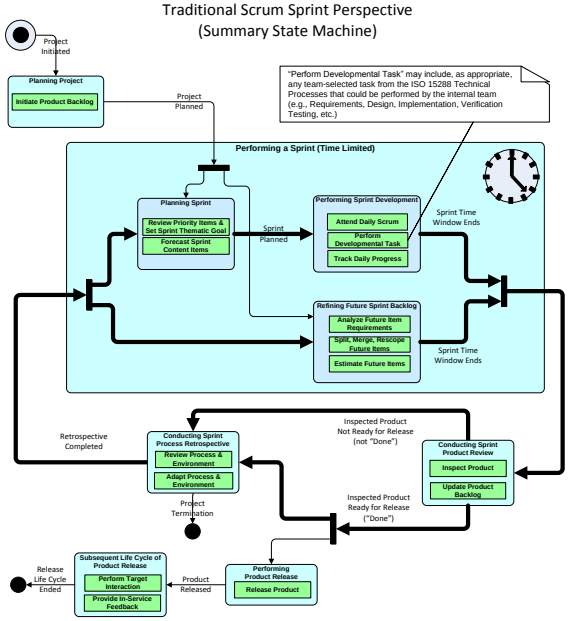
Performs most of these ISO processes, to manage what is being learned about S2 space.

ASELCM Pattern: System 1 (Target System) agility, driven by System 2 (Life Cycle Management) and System 3 (System of Innovation).

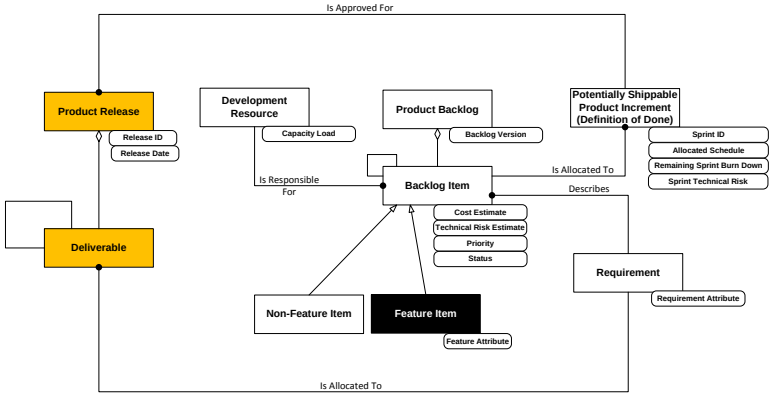
Agile Scrum Model

(See IW2015 MBSE Workshop Attachment I for more.)

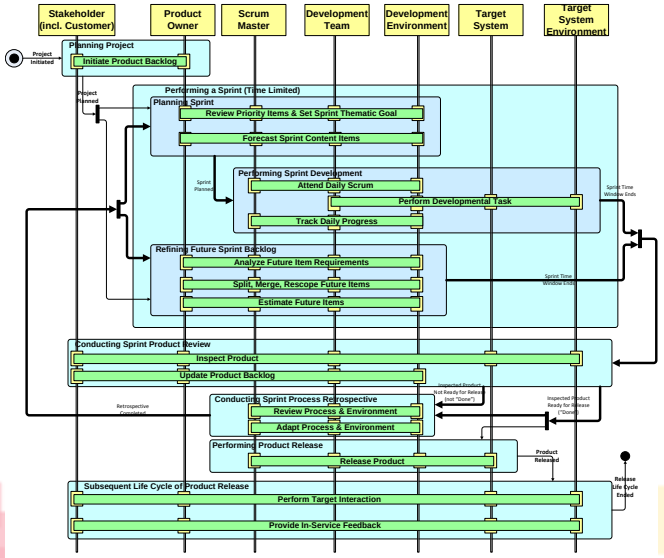
Traditional Scrum Sprint Perspective
(Summary State Machine)



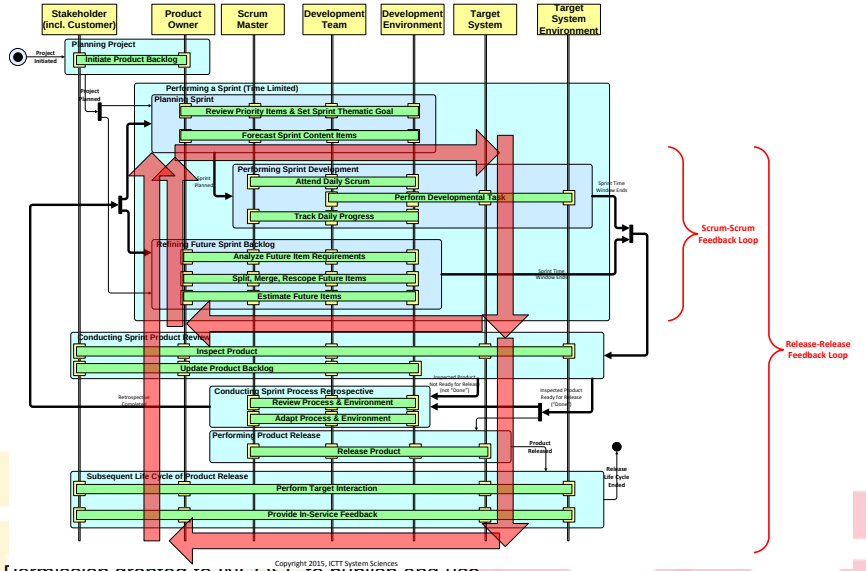
Traditional Scrum Sprint Perspective
(Simplified Model of Managed Information)



Traditional Scrum Sprint Perspective
(Activity Diagram, with Swim Lane Roles)



Traditional Scrum Sprint Perspective
(Activity Diagram, with Swim Lane Roles)

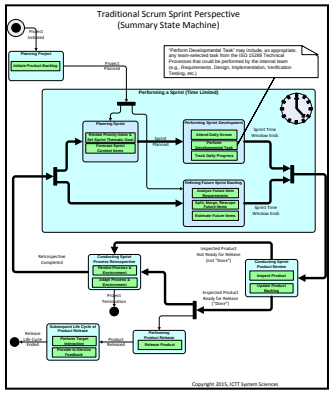


More Than One Representation (Model View) of the Same Underlying Reality

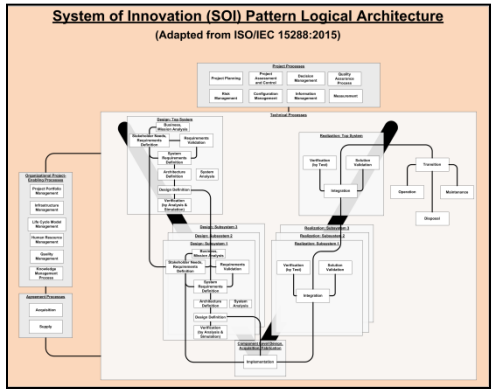
We are dealing with four different representations of the same underlying reality:

1. The Scrum Pattern: Emphasizes time-bound outputs and feedback, focusing on processes for *learning from produced outputs, and management of risk*
2. The ISO15288 Pattern: Emphasizes types of processes, focusing on *management of processes*
3. The Agile Systems Engineering Life Cycle Pattern : Shows how (1) and (2) above may be seen as one
4. The S*Metamodel: Emphasizes the information flowing through all three of them: (1), (2), and (3)

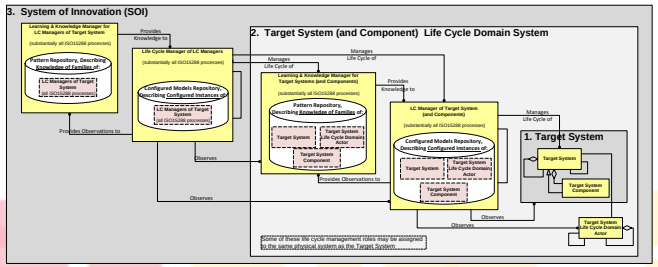
Scrum Pattern



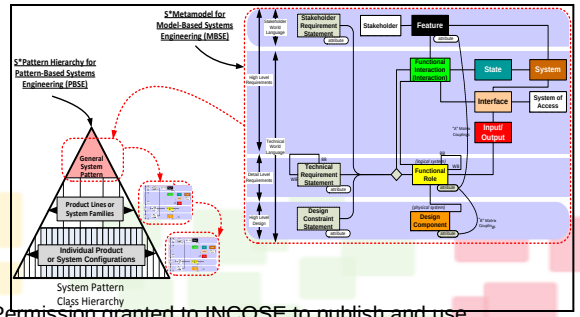
ISO15288 Pattern



ASELC Pattern



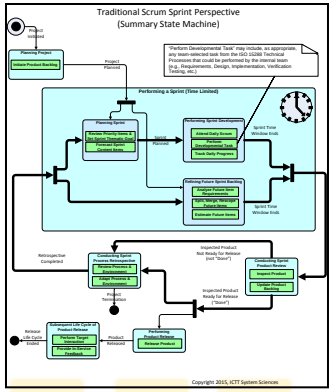
S*Metamodel

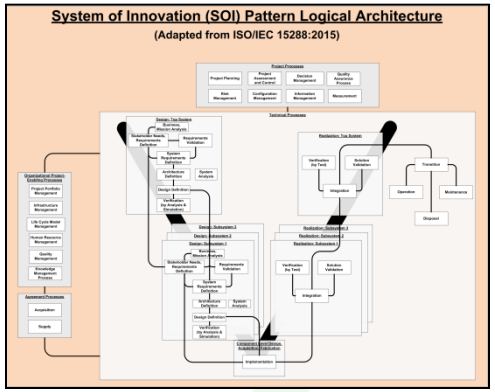


More Than One Representation (Model View) of the Same Underlying Reality

- The Scrum Model is actually an abstraction of the more complex-looking multiple Processes of the ISO15288 System Life Cycle reference model:
  - As indicated in the Agile literature, nothing about the Scrum Model is intended to prevent things like Requirements Analysis, Verification (Test), or even aspects of Project Management, . . .
  - But those activities are shared by the small team members who play many individual roles, and the simpler-looking Scrum model "gives us permission" to "do what is needed" in a given situation, in an "agile way".

Scrum Pattern





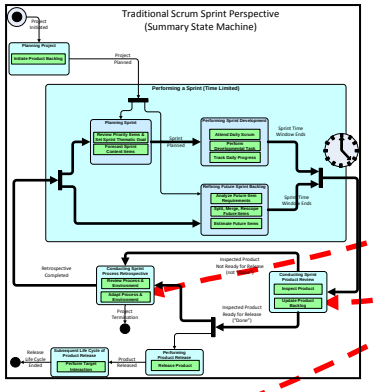
ISO15288 Pattern

More Than One Representation (Model View) of the Same Underlying Reality

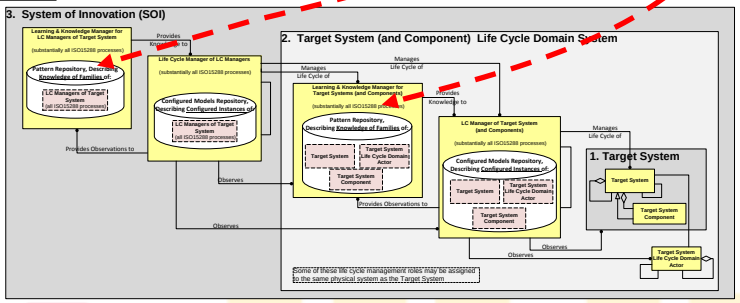
- The Scrum Model also abstracts complex learning behavior, into simple-looking form—but it is still strongly expected to occur as part of the Agile Process, and is more explicitly represented in the ASELC Pattern, as capture of Pattern information—not assumed to be only in human minds.

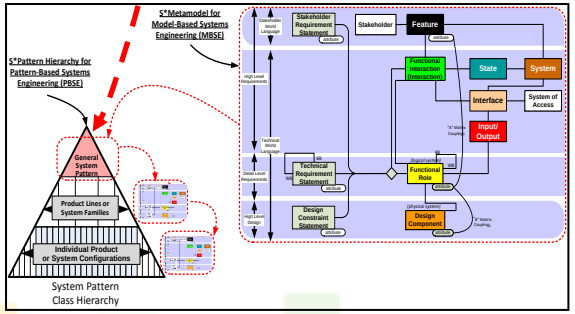
Learning

Scrum Pattern



ASELC Pattern





S*Metamodel

Learning often in upper-most S1,2,3 Pattern, but can also be in specializations and configurations below it.

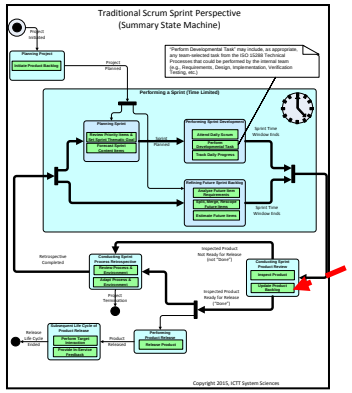
More Than One Representation (Model View) of the Same Underlying Reality

- Notice that the division of the System 2&3 learning roles in the ASELCM Pattern corresponds to the Scrum division of (review and learning about target system) versus (review and learning about development process):

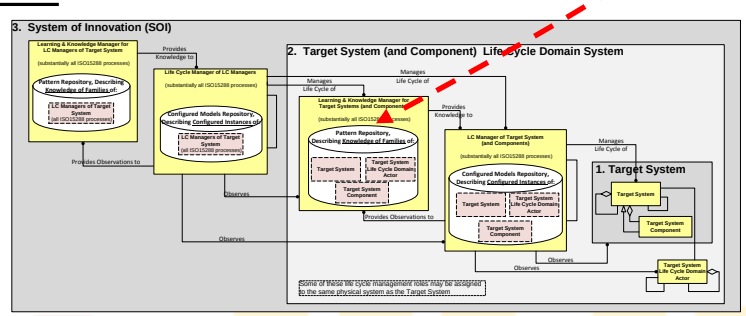
Pattern: Learnings about Target System (Product & Its Environment)

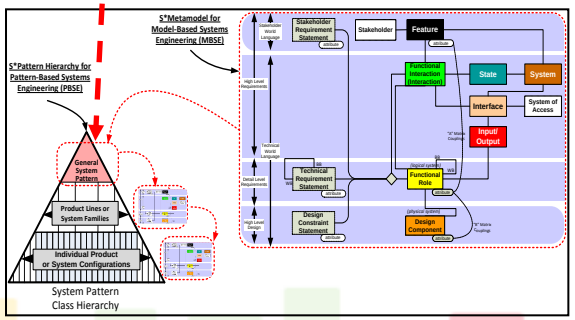
Pattern: Learnings about Development / Fielding System & Its Environment

Scrum Pattern



ASELC Pattern





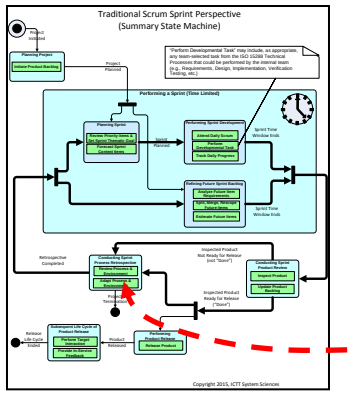
S*Metamodel

More Than One Representation (Model View) of the Same Underlying Reality

- Notice that the division of the System 2&3 learning roles in the ASELCM Pattern corresponds to the Scrum division of (review and learning about target system) versus (review and learning about development process):

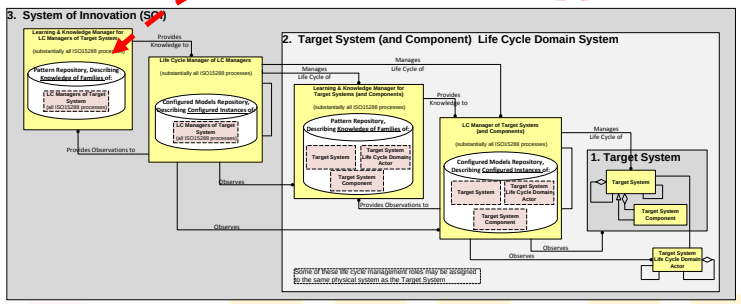
Pattern: Learnings about Target System (Product & Its Environment)

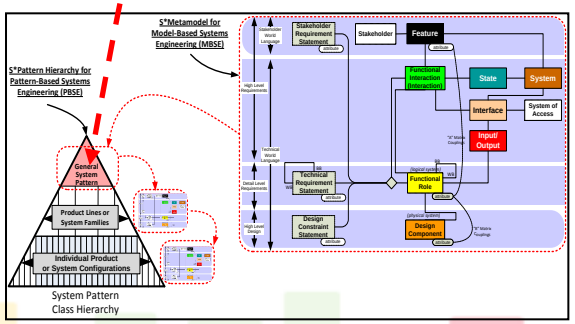
Scrum Pattern



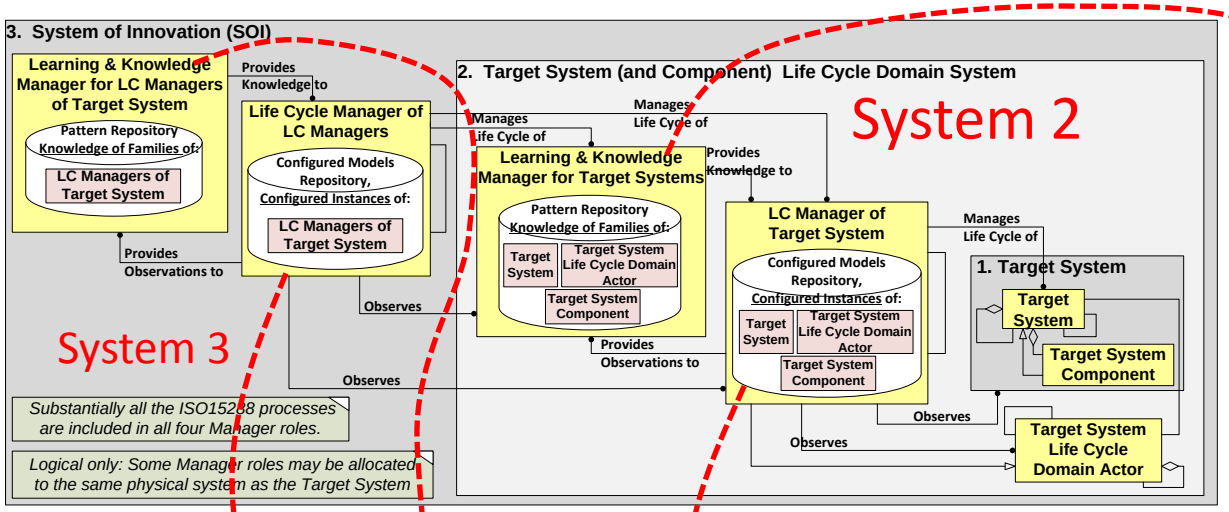
Pattern: Learnings about Development / Fielding LC Management System(s) & Its Environment

ASELC Pattern

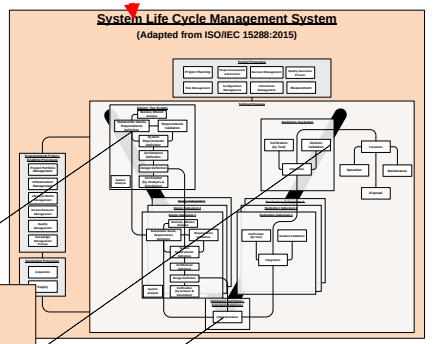




S*Metamodel

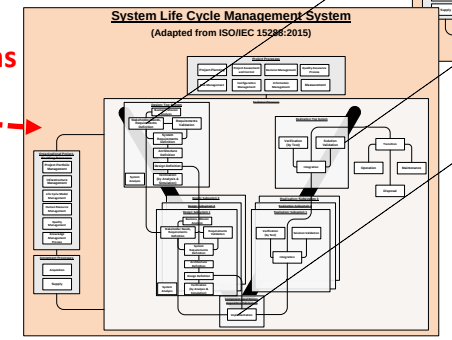


Performs most of these ISO processes, to manage what is being learned about S1 space.

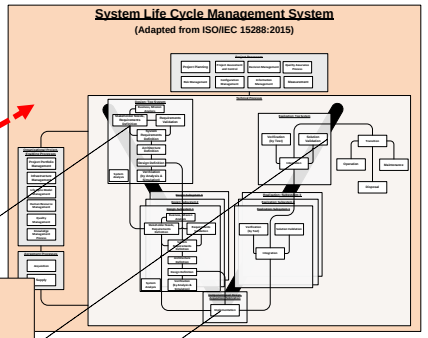


Manages Target System Family Patterns

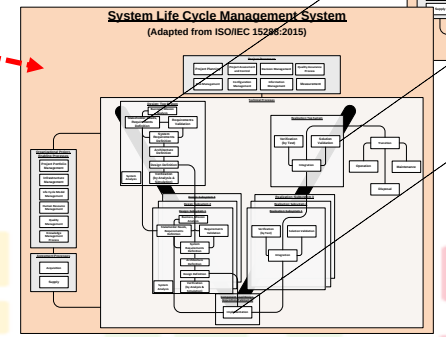
Performs most of these ISO processes, to manage S1 instances , using S1 patterns



Manages LC Management System Family Patterns



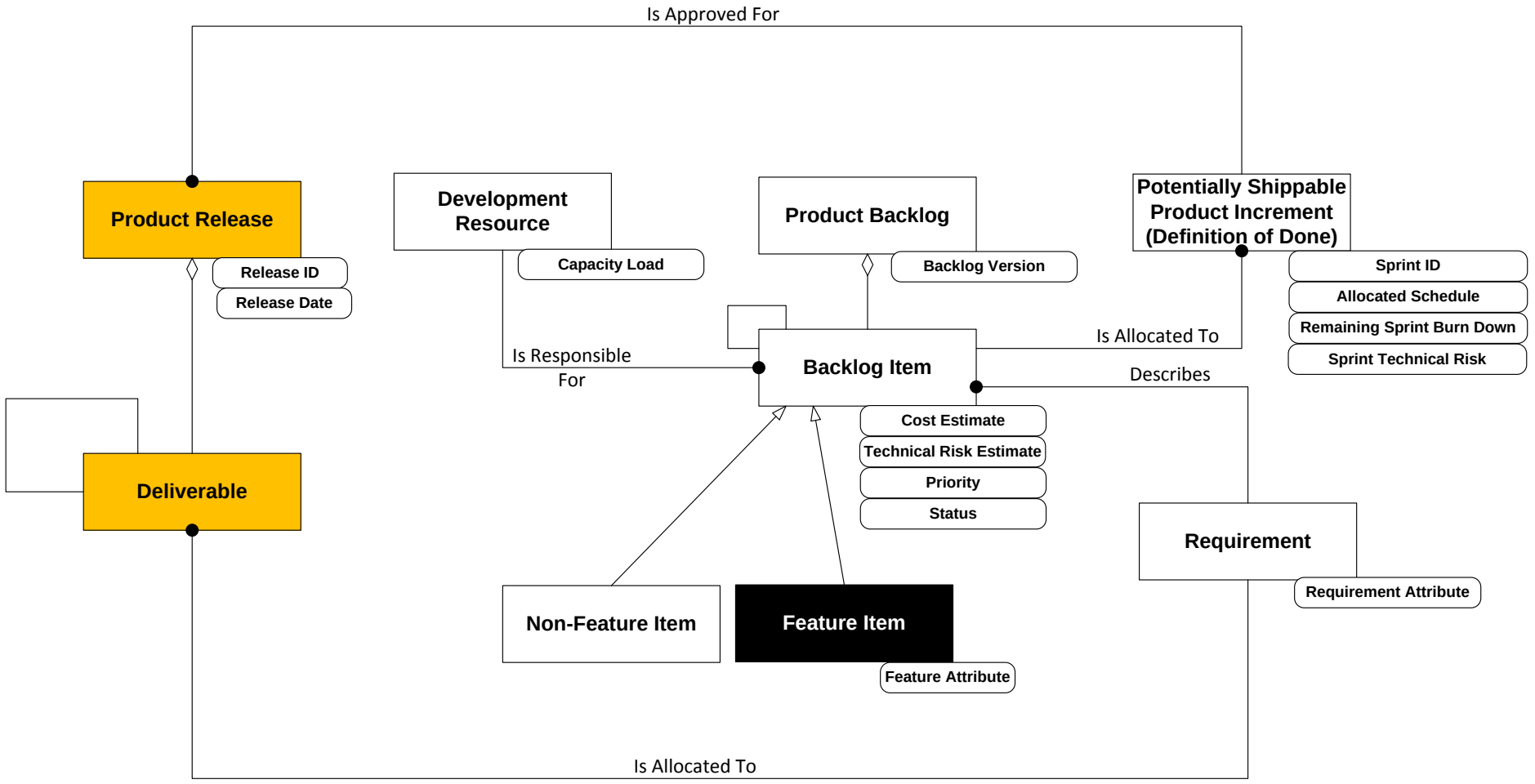
Performs most of these ISO processes, to manage S2 instances , using S2 patterns



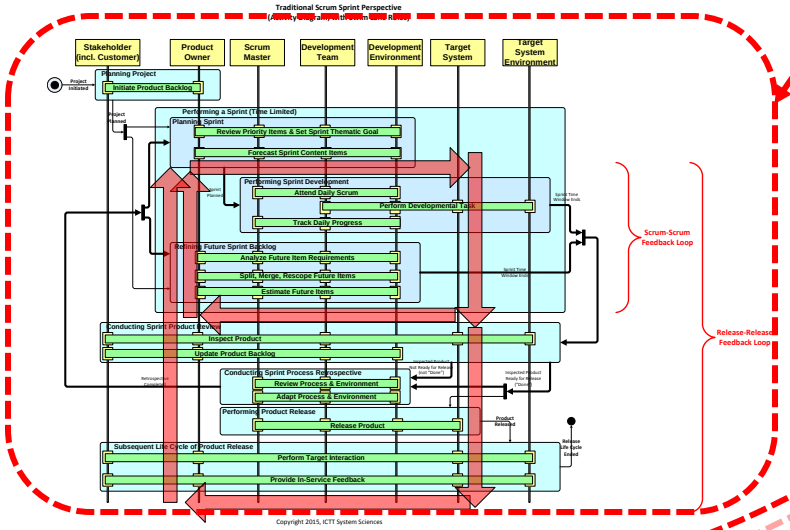
Performs most of these ISO processes, to manage what is being learned about S2 space.

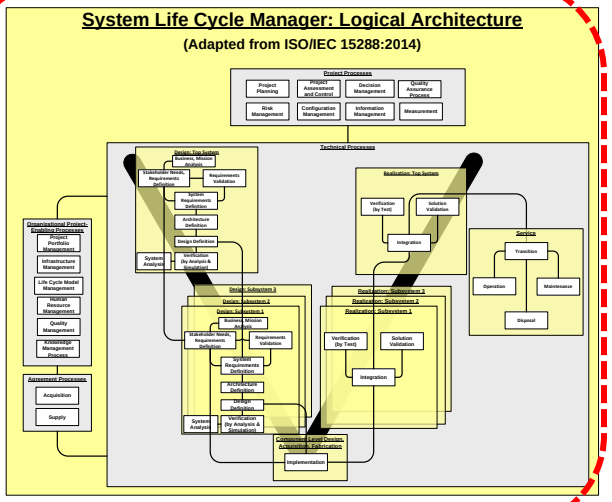
ASELCM Pattern: System 1 (Target System) agility, driven by System 2 (Life Cycle Management) and System 3 (System of Innovation).

Traditional Scrum Sprint Perspective
(Simplified Model of Managed Information)



Agile Scrum Specialization of the ASELCM Pattern





4 Instances