25th anniversary
annual INCOSE
international workshop
Los Angeles, CA
January 24 - 27, 2015

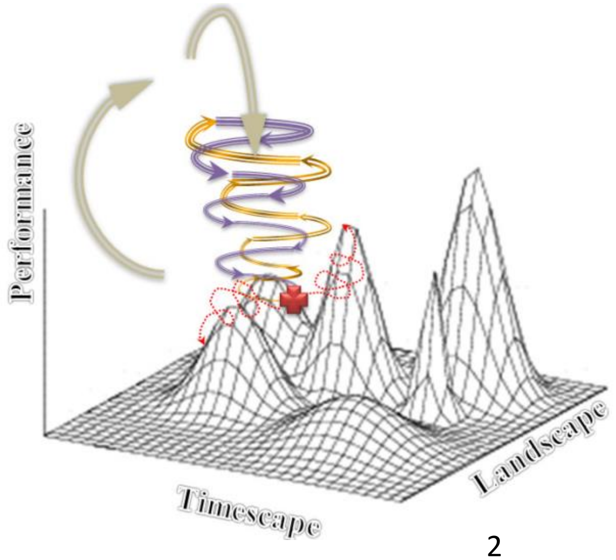
Session Presentation : Introduction to the Agile Systems Pattern

An MBSE-Based System Pattern, with Implications for Agile Modeling

Bill Schindel, ICTT System Sciences

schindel@ictt.com

Contents



Objectives of Overall Breakout Session on Agile Modeling and Modeling Agile Systems

- Gain a broader understanding of Agile Systems-engineering and the Engineering of Agile-Systems, and the common fundamental Agile Architecture Pattern of both that enables effective response in uncertain, unpredictable, and evolving SE and operational environments.
- Learn how formal model-based System Patterns can expand the fundamental Agile Architecture Pattern with necessary agile-enabling details for fleshing out an agile SE process and agile system design.
- Understand the role and impact of accumulated system patterns within Agile Systems.
- Learn about Pattern-Based Systems Engineering (PBSE), and how Agile Modeling is facilitated by PBSE.
- Learn how S*Patterns express model-based system patterns.
- Find out about the 2015/16 traveling workshop Agile System Engineering Life Cycle Model (ASELCM) Fundamentals project, to occur in the US and Europe, along with how and why you and your organization might want to participate

<u>Assumption</u>: Some awareness of the general ideas of Agile Systems from this breakout's preceding sub-session, or the listed References

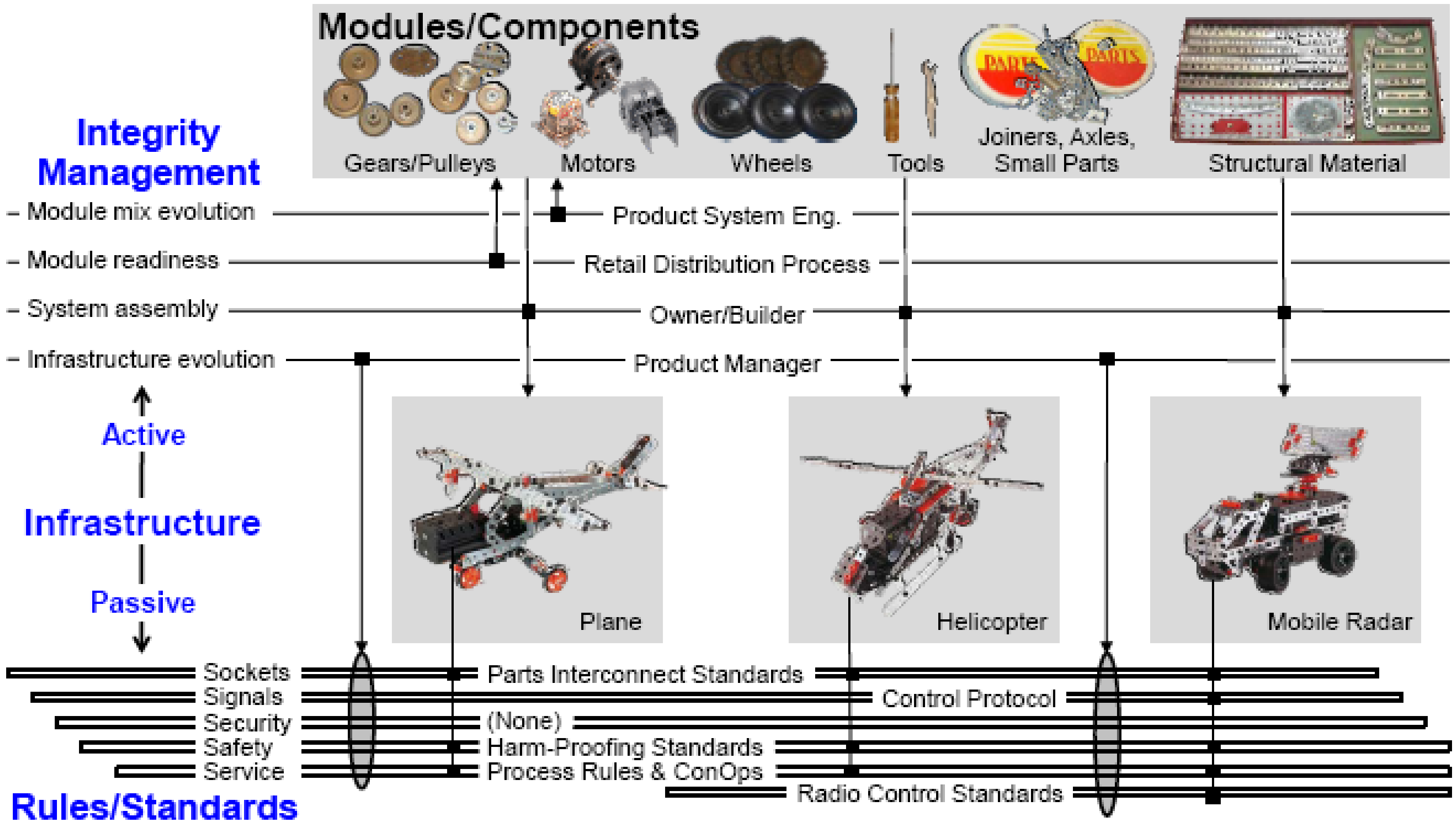
Goals of this sub-session

- Review a summary of major segments of the Agile Systems Pattern.
- From this example, learn about model-based, reusable, configurable representations of system patterns using the S*Metamodel.
- Understand the implications of the "Experience Accumulation" subsystem of the Agile Systems Pattern, for Agile Modeling, as well as other implications.
- Find out how to learn more.

A Peek Ahead

The Agile System Pattern will capture (in S*Model) the key ideas associated with the pre-MBSE Agile System Architecture:

- As in (Dove and LaBarge, 2014)



A Peek Ahead

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

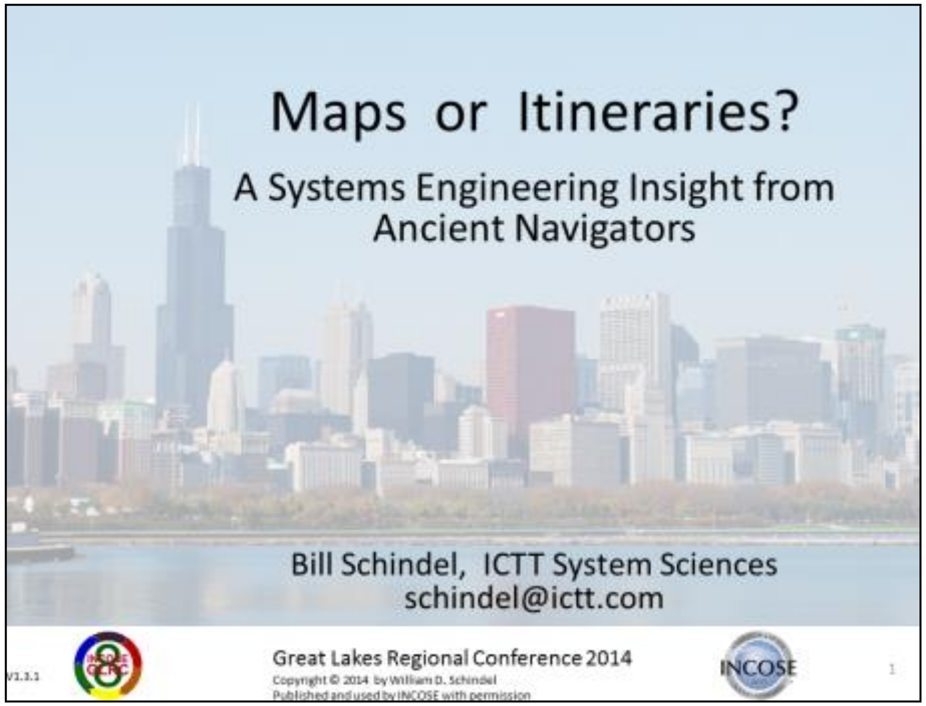
Vital for Scrum, other approaches

Vital for Response Situation Analysis (RSA)

(ed: see also notes at end of slides, on anticipations) (also coordinate with implications section)

Quick Digest of Two Key Perspectives

- In addition to Agile Systems base concepts
- Two key perspectives:
  - Maps versus Itineraries: SE Information *vs*. SE Process
  - System Life Cycle Trajectories in S*Space



Maps or Itineraries?

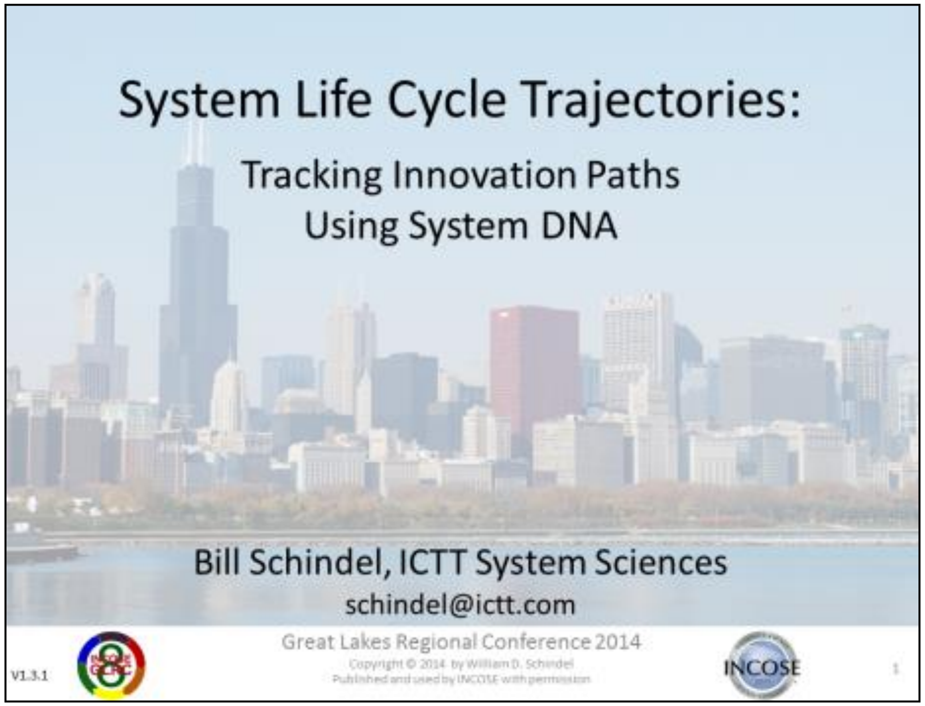
A Systems Engineering Insight from Ancient Navigators

Bill Schindel, ICTT System Sciences
schindel@ictt.com

V1.3.1

Great Lakes Regional Conference 2014
Copyright © 2014 by William D. Schindel
Published and used by INCOSE with permission

1



System Life Cycle Trajectories:

Tracking Innovation Paths Using System DNA

Bill Schindel, ICTT System Sciences
schindel@ictt.com

Great Lakes Regional Conference 2014
Copyright © 2014 by William D. Schindel
Published and used by INCOSE with permission

V1.3.1

1

(See references)

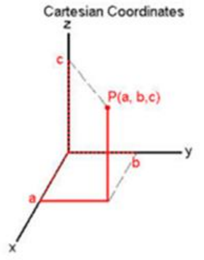
Maps *vs*. Itineraries -- SE Information *vs*. SE Process

Itinerary ≠ Map!

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

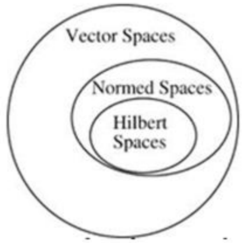
When they eventually did emerge, maps represented a newer idea of the nature of "where".

- The SE Process consumes and produces <u>information</u>.
- But, SE historically emphasizes <u>process</u> over <u>information</u>. (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 <u>*vs*</u>. 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

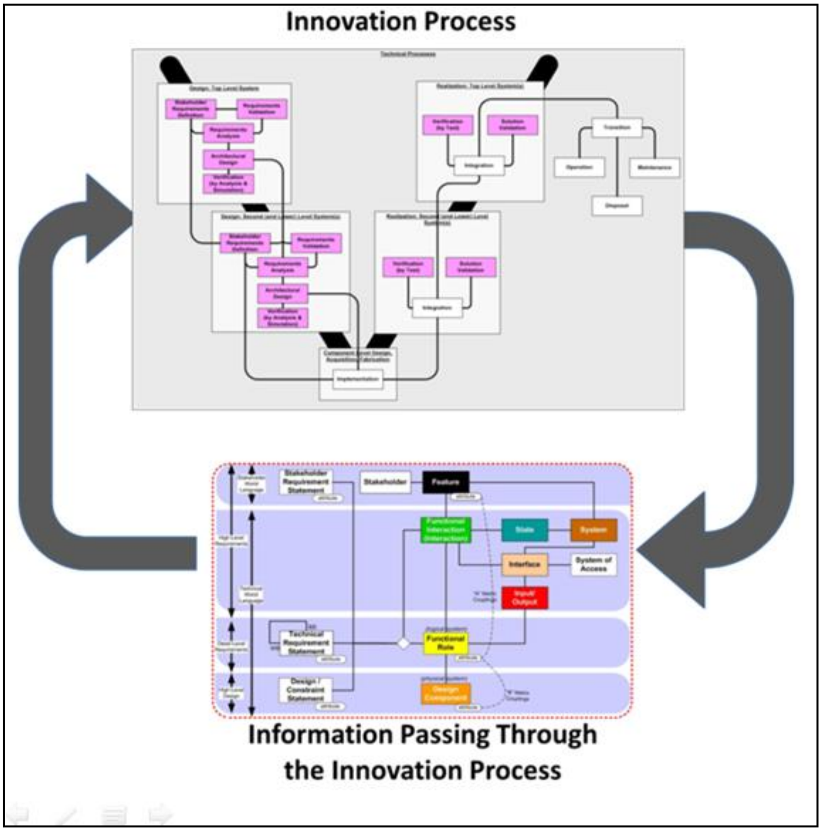
Geometrization of Algebra, by Rene Descartes



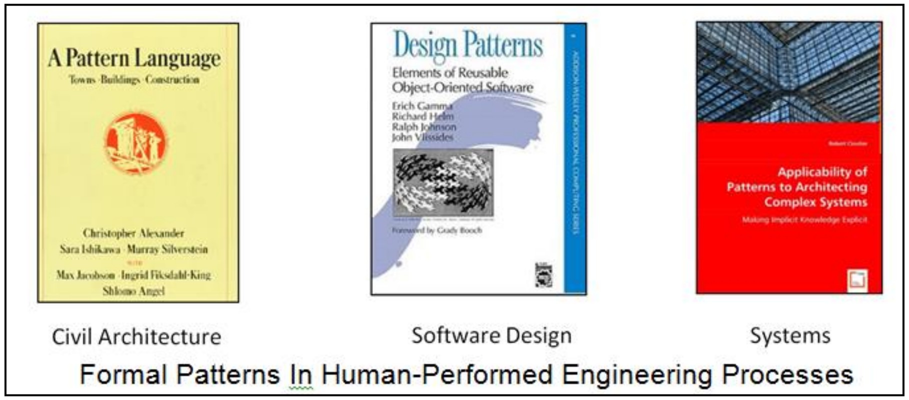
David Hilbert
1862 - 1943

Geometrization of Function Space, by David Hilbert

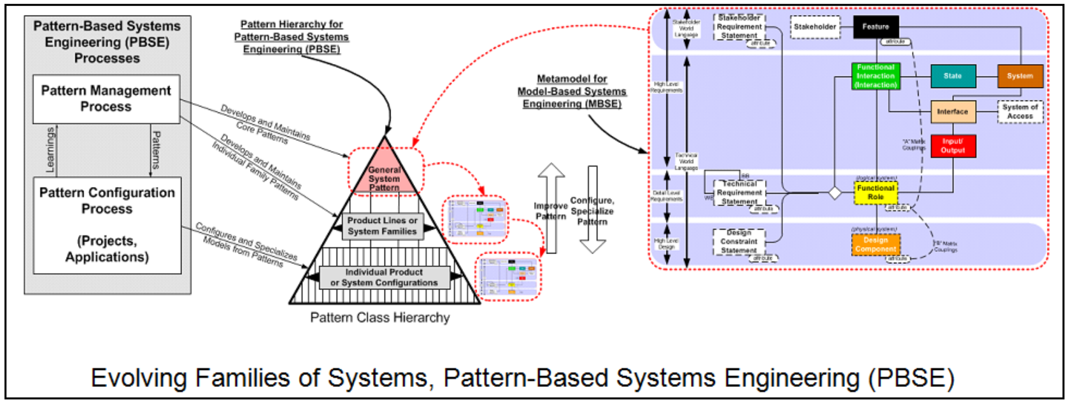
Maps *vs.* Itineraries -- SE Information *vs.* SE Process



Innovation Process
Technical Processes
Design, Top Level Systems
Realization, Top Level System(s)
Stakeholder Requirements Definition
Requirements Validation
Requirements Analysis
Architectural Design
Verification (by Analysis & Simulation)
Verification (by Test)
Solution Validation
Integration
Transition
Operation
Maintenance
Disposal
Design, Second (and Lower) Level Systems
Realization, Second (and Lower) Level Systems
Component Level Design, Acquisition, Fabrication
Implementation
Stakeholder Requirement Statement
Stakeholder
Feature
Functional Interaction (Interaction)
State
System
Interface
System of Access
Input/Output
Technical Requirement Statement
Functional Role
Design Constraint Statement
Design Component
Information Passing Through the Innovation Process



A Pattern Language
Towns · Buildings · Construction
Christopher Alexander
Sara Ishikawa · Murray Silverstein
Max Jacobson · Ingrid Fiksdahl-King
Shlomo Angel
Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
Applicability of Patterns to Architecting Complex Systems
Making Implicit Knowledge Explicit
Civil Architecture
Software Design
Systems
Formal Patterns In Human-Performed Engineering Processes



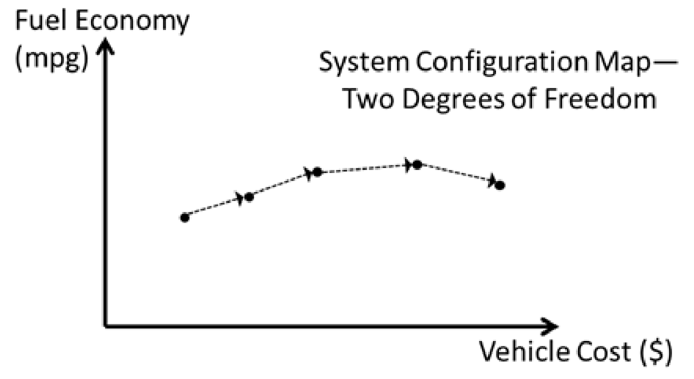
Pattern-Based Systems Engineering (PBSE) Processes
Pattern Management Process
Learnings
Patterns
Pattern Configuration Process
(Projects, Applications)
Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
Develops and Maintains Core Patterns
Develops and Maintains Individual Family Patterns
Configures and Specializes Models from Patterns
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
Pattern Class Hierarchy
Metamodel for Model-Based Systems Engineering (MBSE)
Improve Pattern
Configure, Specialize Pattern
Stakeholder Requirement Statement
Stakeholder
Feature
Functional Interaction (Interaction)
State
System
Interface
System of Access
Input/Output
Technical Requirement Statement
Functional Role
Design Constraint Statement
Design Component
Evolving Families of Systems, Pattern-Based Systems Engineering (PBSE)

- 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 15288.

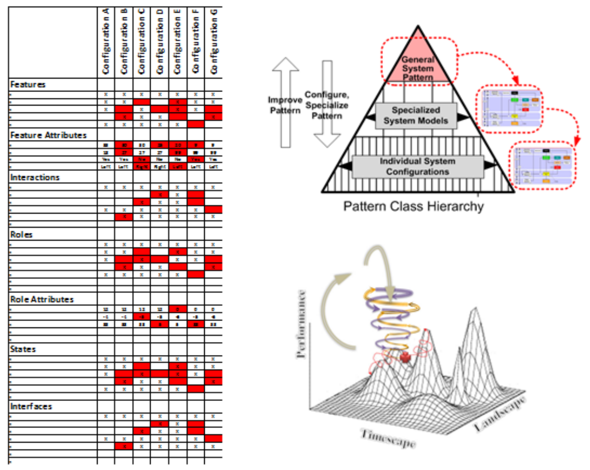
System Life Cycle Trajectories in S*Space

- 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

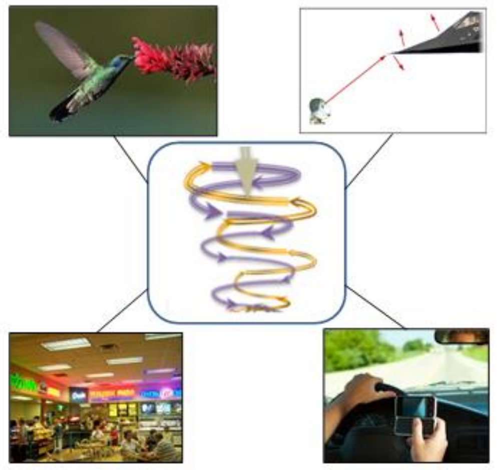
Evolving Systems Over Multiple Life Cycles



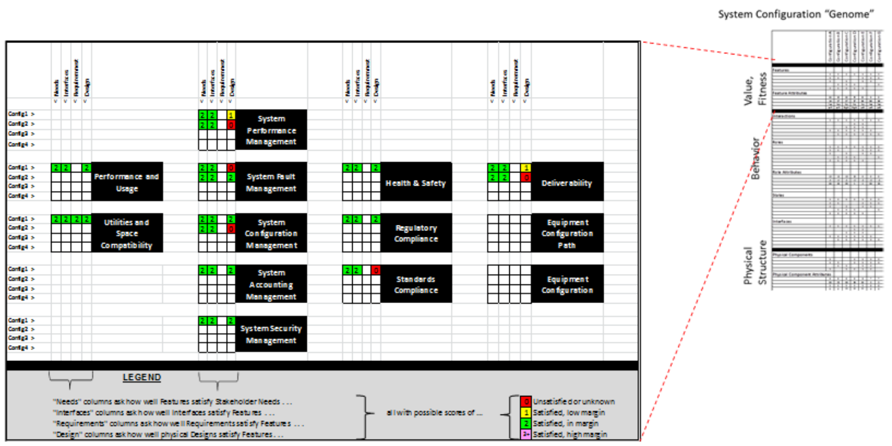
Path as a series of system configurations, through iterations of the SE process



"Delta" Descriptions Further Compress Trajectory Representations


Co-Evolution of Interacting Systems

System Life Cycle Trajectories in S*Space



"Needs" columns ask how well Features satisfy Stakeholder Needs . . .
"Interfaces" columns ask how well Interfaces satisfy Features . . .
"Requirements" columns ask how well Requirements satisfy Features . . .
"Design" columns ask how well physical Designs satisfy Features. . .

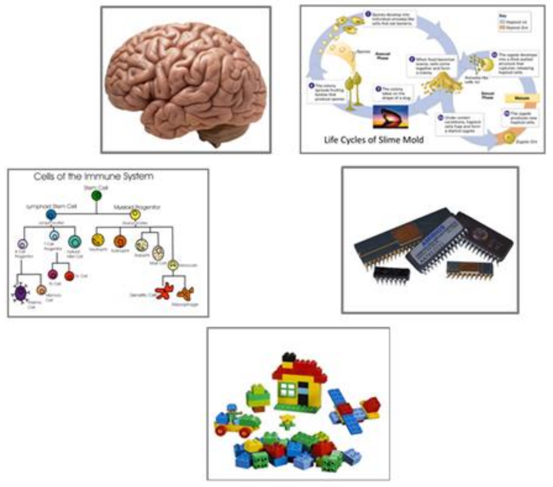
all with possible scores of . . .

0 Unsatisfied or unknown
1 Satisfied, low margin
2 Satisfied, in margin
2+ Satisfied, high margin

A View of the S*Stakeholder Feature Subspace Status

Four Different Configuration Times During System Life Cycles

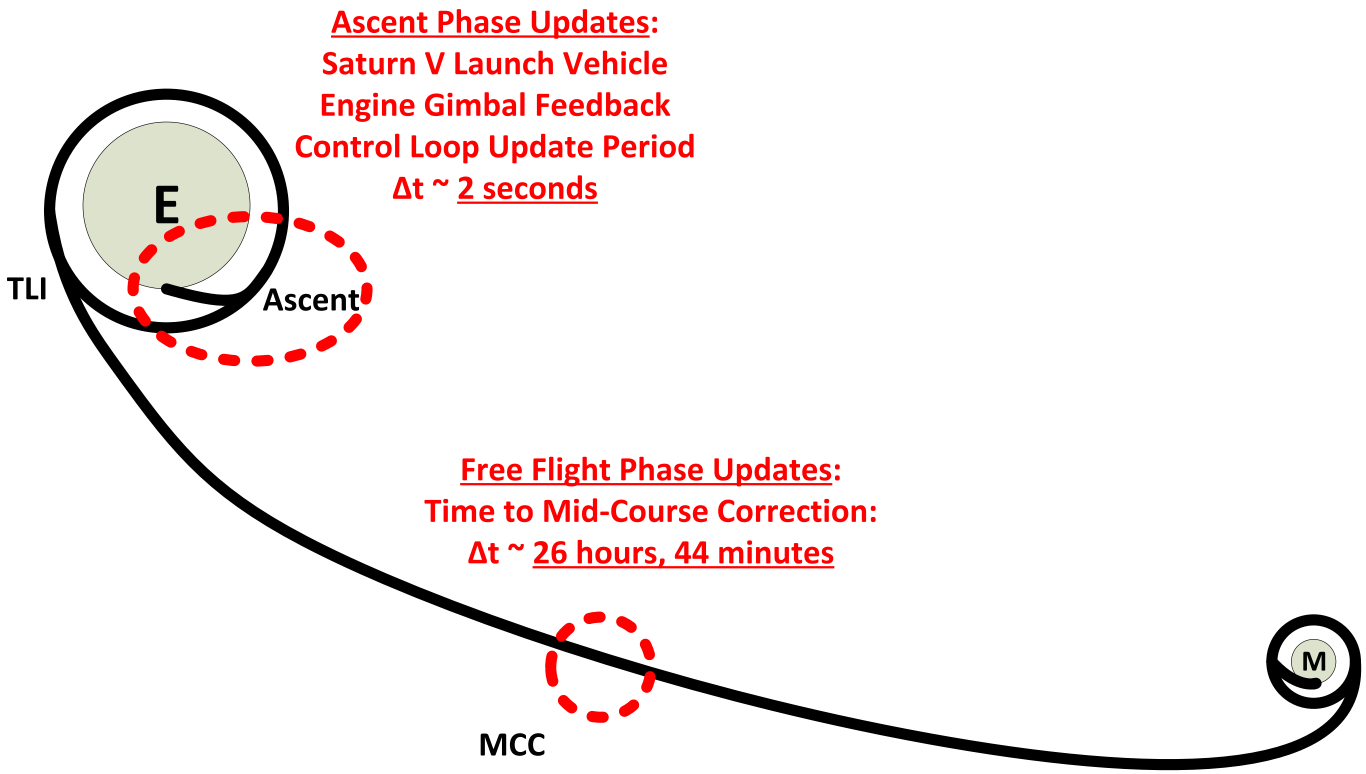
- There are productive "views" of those trajectories, which may be implemented on most any general systems modeling tool or PLM system:
  - a risk management application of SE tracing—
  - projecting detected gaps onto Stakeholder Feature space to understand their significance.
  - productive "views" help know "where" we are, and manage trajectory direction, critical to scrum empiricism
- Progressive advances in configurability:
  - Deferred times of reconfiguration (but S*Space applies to all of them!)
  - Addition of information to architecture
  - Composable architecture



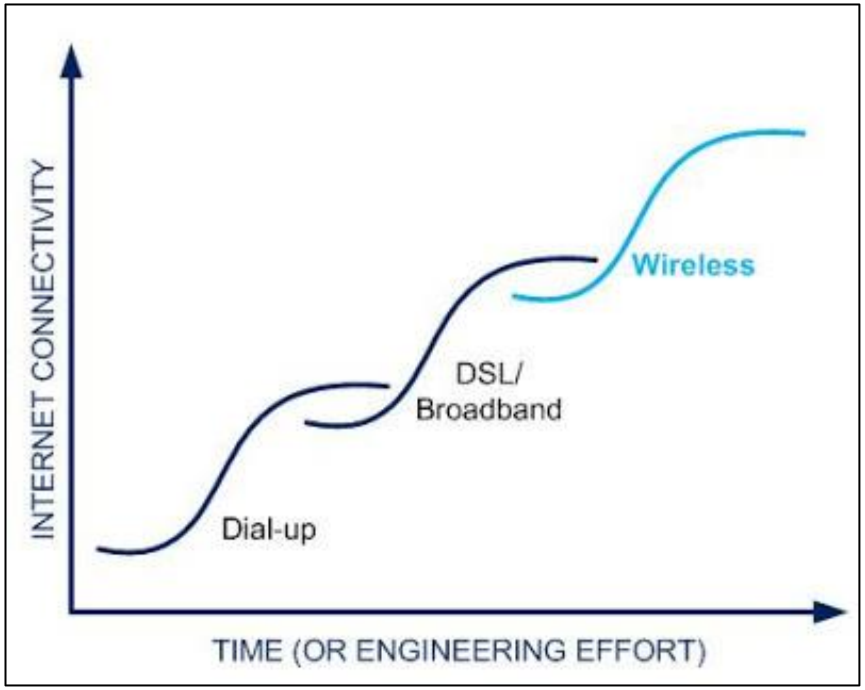
Composable Systems and Component Libraries

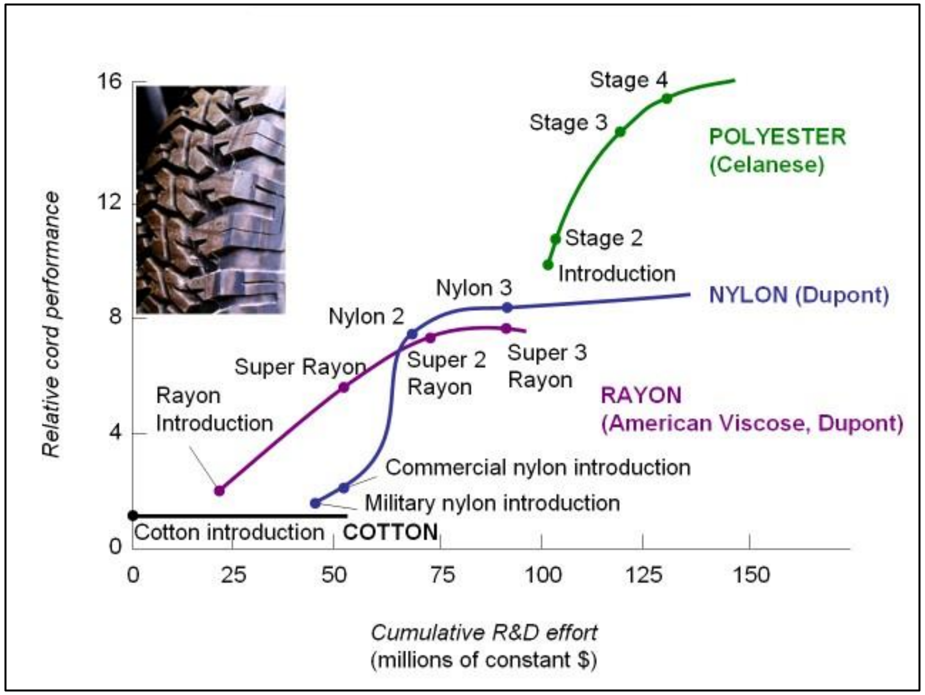
Feedback & Correction Cycle Rate: A Hallmark of Agile Methods

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? [42,43]


Ascent Phase Updates:
Saturn V Launch Vehicle
Engine Gimbal Feedback
Control Loop Update Period
Δt ~ 2 seconds
E
TLI
Ascent
Free Flight Phase Updates:
Time to Mid-Course Correction:
Δt ~ 26 hours, 44 minutes
MCC
M

Additional Trajectory-Based Concept: Learning Curves, "S" Curves

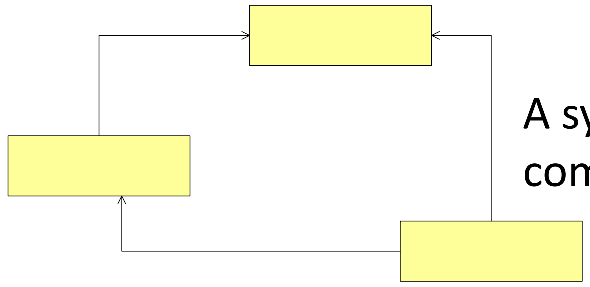




(insert references)

The Agile System Domain Model

- This portion of the Agile Pattern identifies system boundaries (scope) and a few main system names:
  - Rick's differentiation of the two systems in Agile Parts 1 and 2 IS2014 papers was important to the following domain model--even though we intend to "make both of those systems agile".
- In this work (and using S*Models), by "system" we mean a set of interacting components.
- By "interact" we mean to exchange energy, forces, mass flows, or information, so that one component changes the state of another:



A system of interacting components

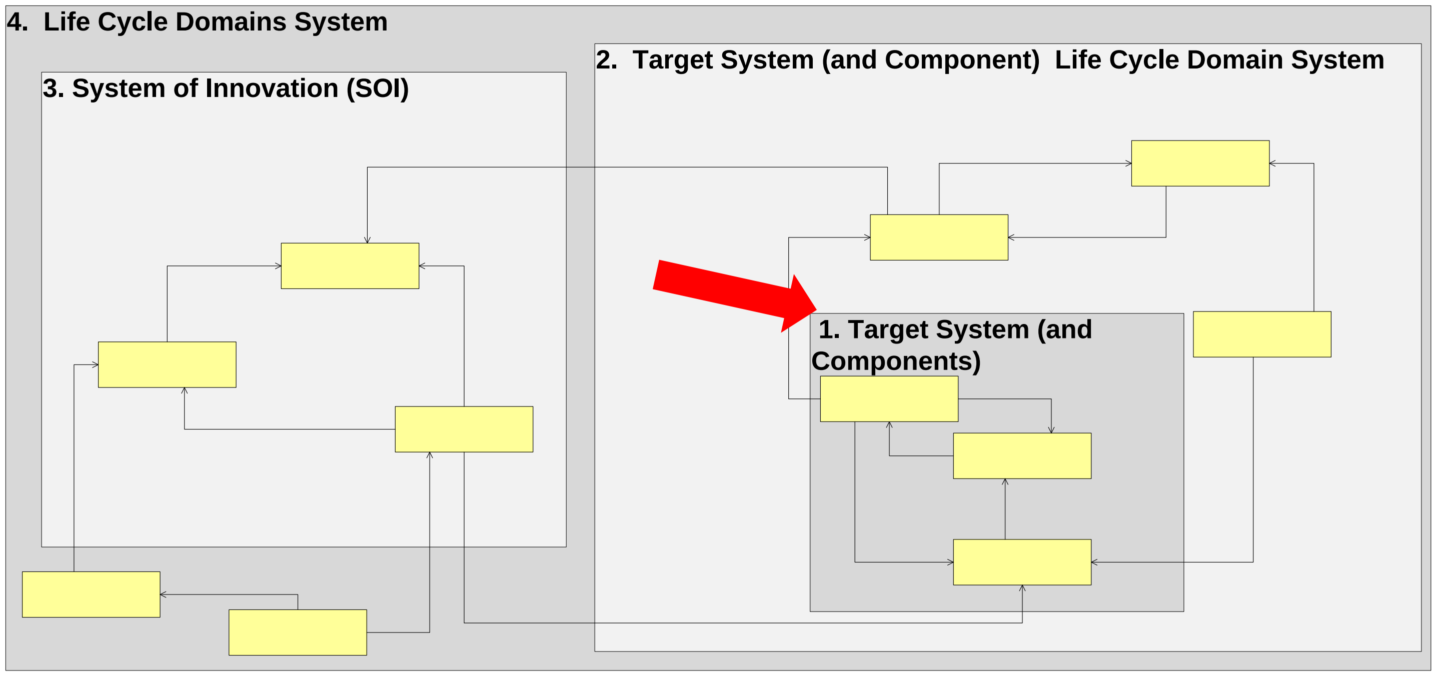
The Agile System Domain Model

- We will particularly refer to **four major system boundaries**:
  - To avoid a confusion bog of loaded terms, we could have just named them “System 1”, “System 2”, “System 3”, and “System 4” 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 **<u>Target System (and Components)</u>:** (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 <u>potentially</u> agile.
- Examples include aircraft, satellites, the human immune system, restaurants, birds, and the health care delivery system.

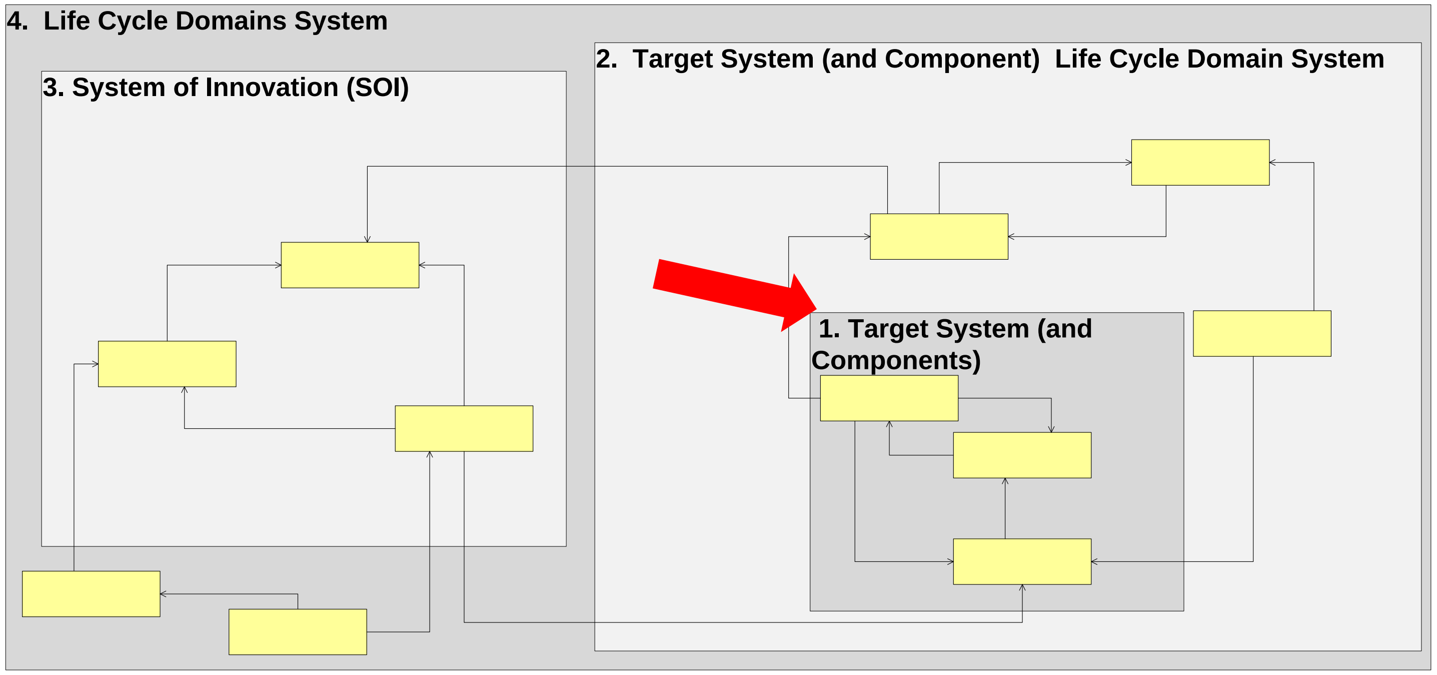

4. Life Cycle Domains System
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)

Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

The Agile System Domain Model

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

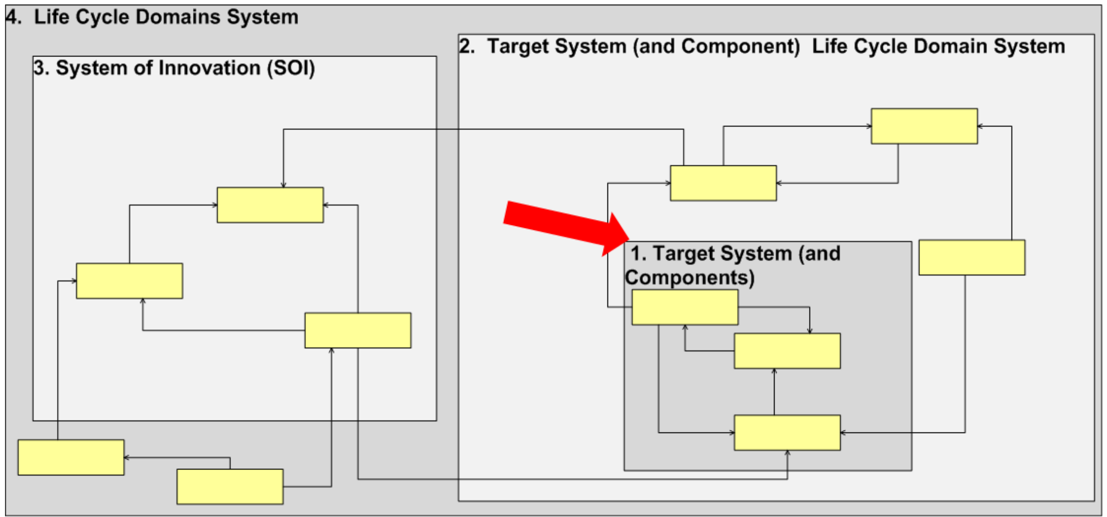
- The Components maintained for integration into a Target System, but not yet integrated, are included in this domain.
- Notice that this idea can apply at multiple additional levels (e.g., SOS, System, Component, etc.)


4. Life Cycle Domains System
2. Target System (and Component) Life Cycle Domain System
3. System of Innovation (SOI)
1. Target System (and Components)

Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

Example Target System (for System 1): Home Entertainment System Example

- Wide Area Transport Media (RF, Cable, DSL, etc.)
- Receivers/Tuners (AM, FM, Satellite, TV, Modem, etc.)
- Recorded Media (Vinyl, Mag Tape, CD, etc.)
- Media Players (Record, CD, Tape, DVD)
- Amplifiers
- Speakers
- Display Media (CRT, Plasma, LCD, OED, etc.)
- Local Transport Media (Wiring, Power Line Carrier, Bluetooth, etc.)
- User Controls (Panel, Specialized Remote, Universal Remote, Smart Phone)

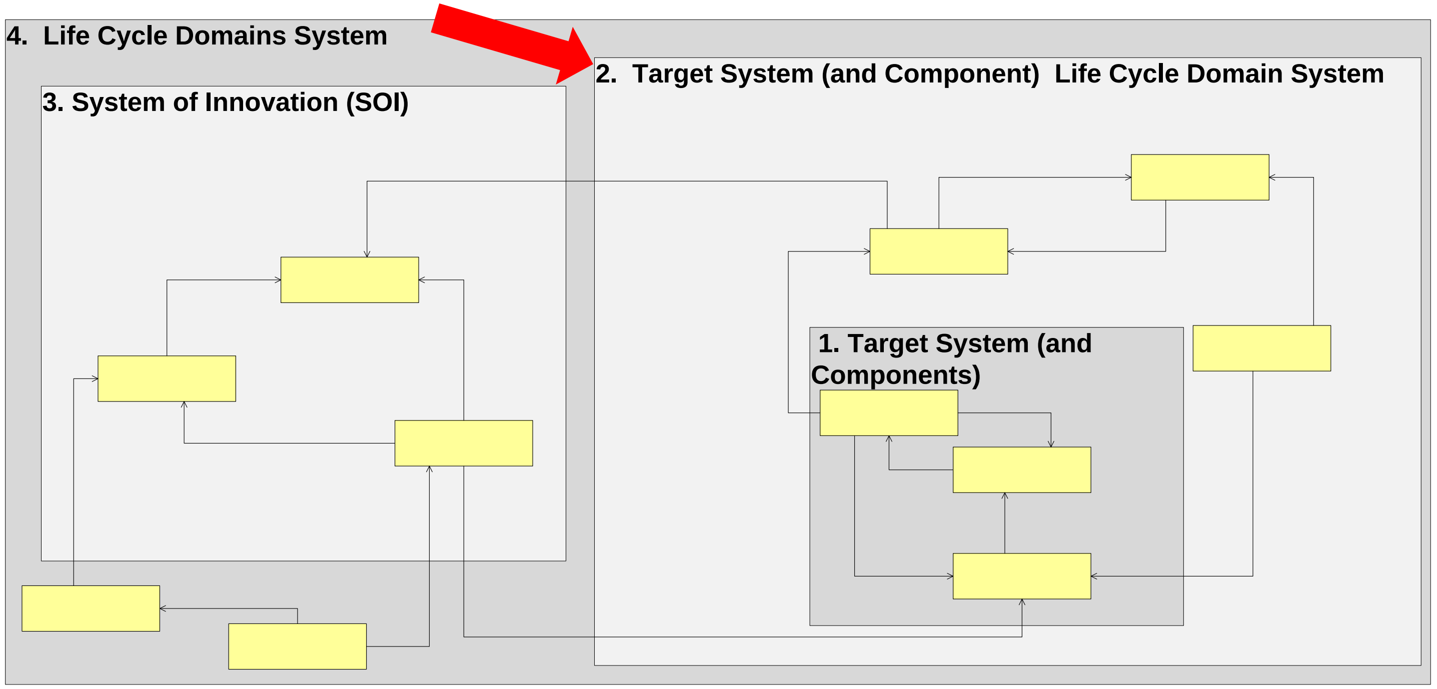

4. Life Cycle Domains System
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)

The Agile System Domain Model

System 2: The **<u>Target System (and Component) Life Cycle Domain System</u>:** (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[1] with which the Target System will directly interact during its life cycle:

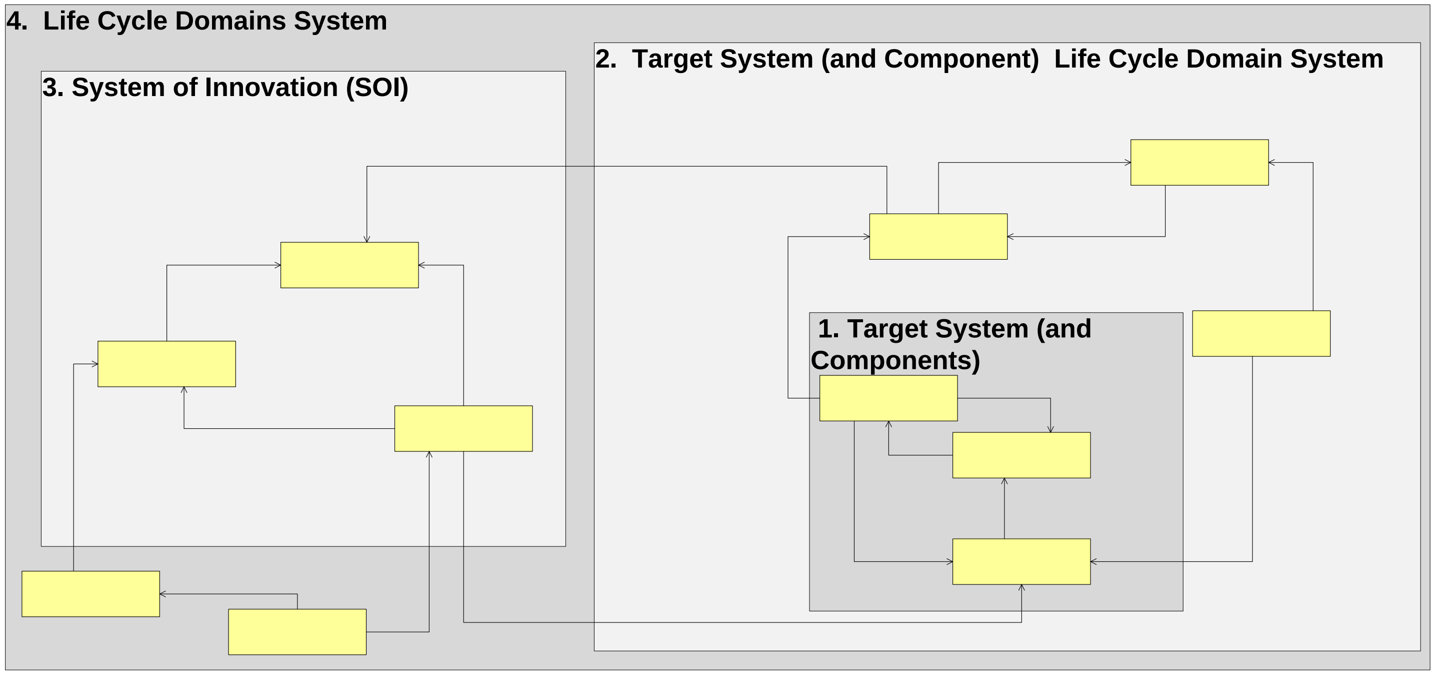
- This includes any system that directly manages the life cycle of an instance of a Target System (or a Component)—production and integration systems, maintenance and operations systems, and others.

1. "Actors" are environmental entities that interact with a system of interest.


4. Life Cycle Domains System
2. Target System (and Component) Life Cycle Domain System
3. System of Innovation (SOI)
1. Target System (and Components)

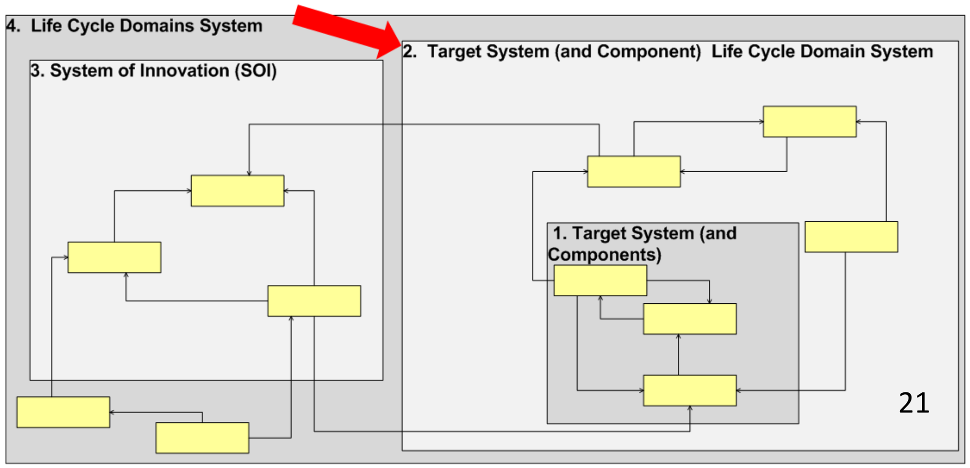
Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

- Again, remember that these are logical (behavioral) roles. In realized physical 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 will add the physical components to the model shortly.


4. Life Cycle Domains System
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)

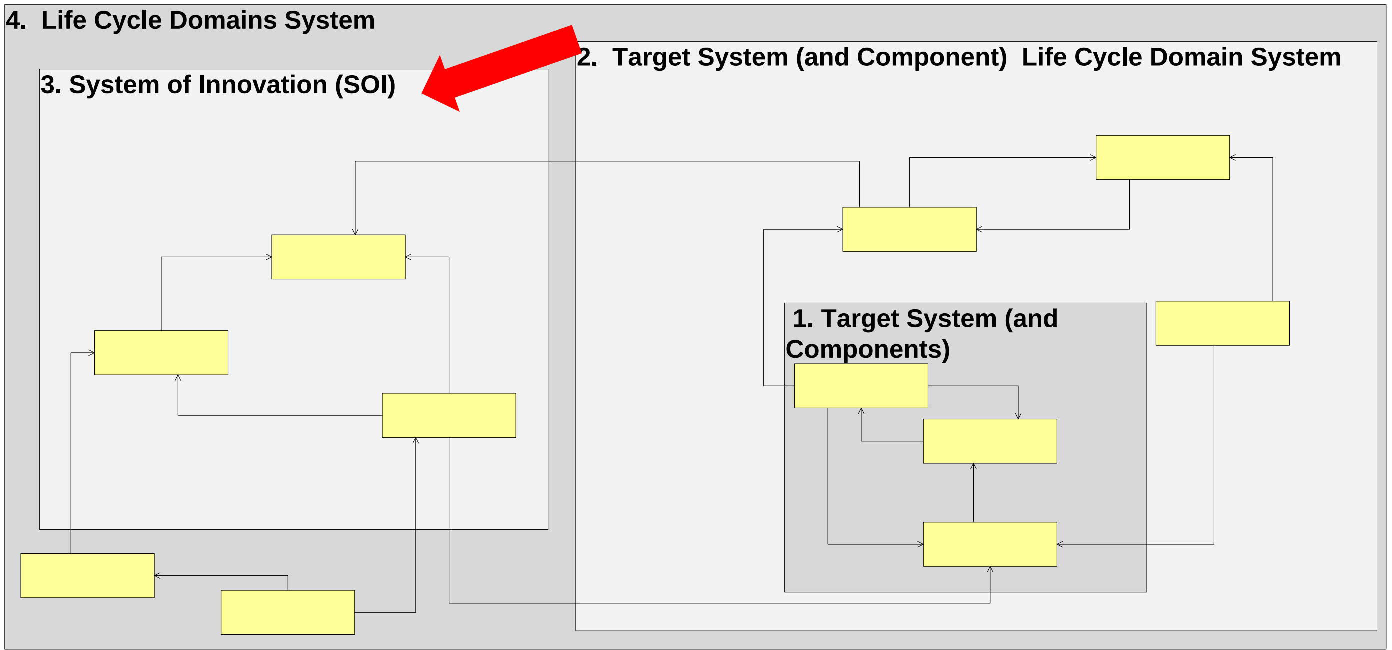
Example Target System (and Component) Life Cycle Domain System (System 2): Home Entertainment System Example

- Supply Chain: Electronics Systems Manufacturer, Distributor, Retailer, Electronic Components Manufacturer
- Operations, Maintenance, Configuration, Performance Management: Home User, Installation Technician, Hand & Electronic Tools, Repair Shop, Manufacturer Warranty Service Center, Manufacturer MES, PLM, CAD Information Systems & Tools
- Security Management: Physical Security, Authentication, Authorization, Encryption
- Other Environmental Actors: Power System, Home Environment, Broadcasters, Media Companies, Content Producers, Content Sellers


4. Life Cycle Domains System
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)

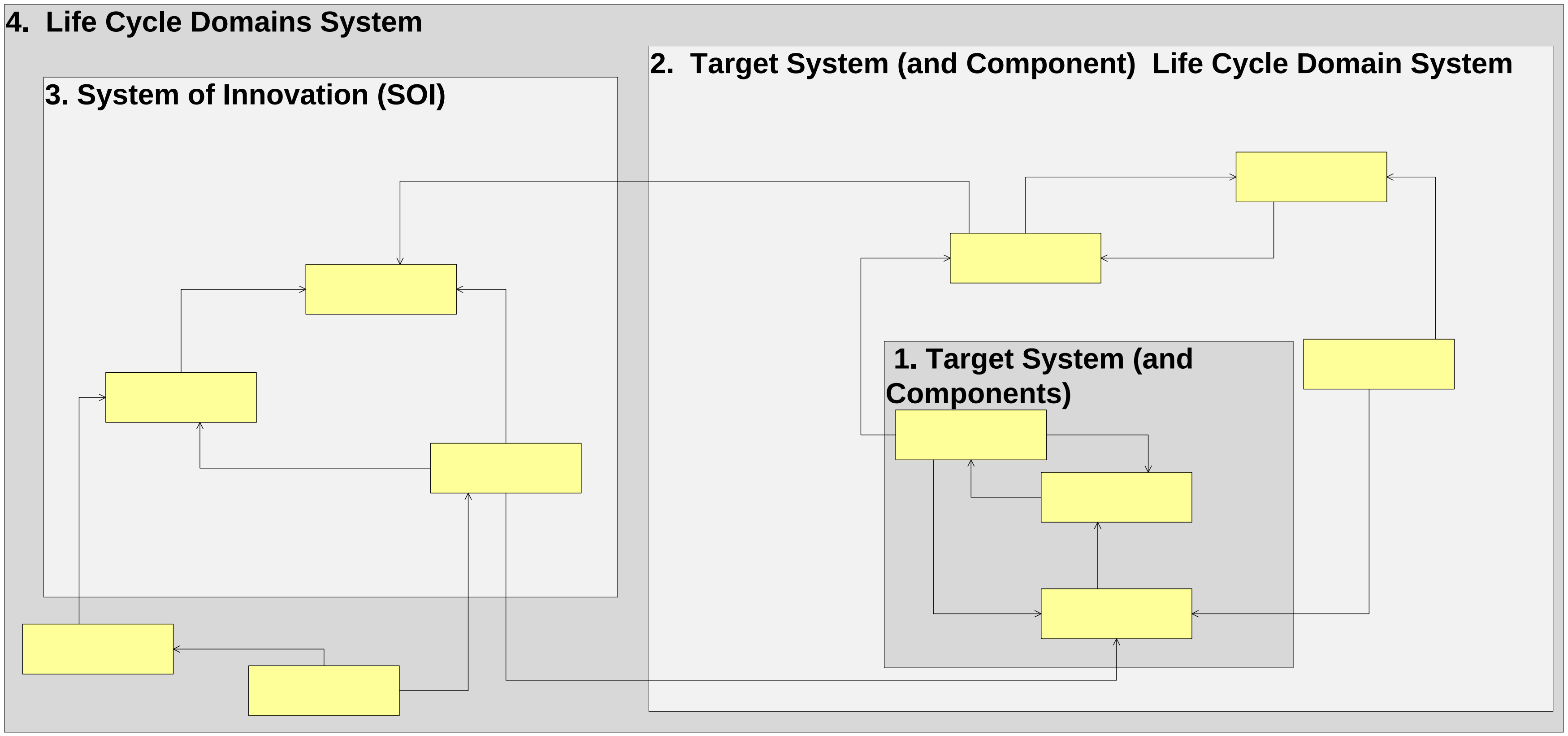
System 3: The **System of Innovation:** The logical system responsible for creating the possibility of (not production of) instances of Target System(s) with new or modified capabilities:

- Includes distillation of new knowledge (by observation) about Target Systems, their life cycle management, and their environmental domains, for future use.
- Also includes creation of instances of new production or other life cycle management capabilities for Target Systems, but not new instances of Target Systems.
- Engineers might think of this as the Engineering Process or the Development Process, but we have given it a more general name--to remind us that an innovation "competitor" may be operating from a cave or kitchen table, lacking a "recognized" engineering process; or, it might be a biological process that did not attend engineering school; or it might be some other type of innovation process, which we will study here.


4. Life Cycle Domains System
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)

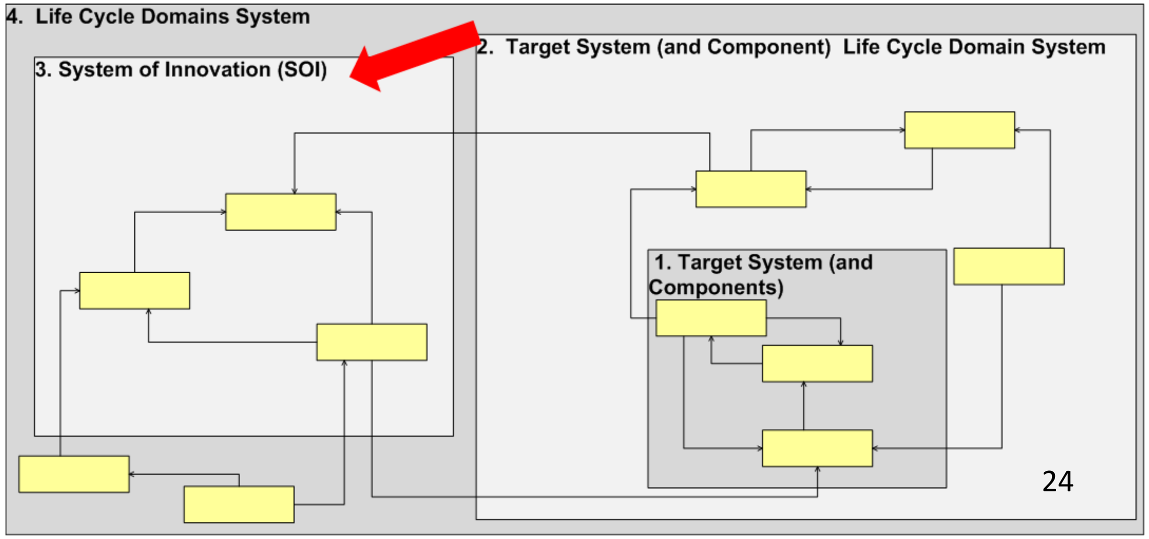
Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

- Summary so far:
  - System 2, the Target System Life Cycle Domain System produces and modifies instances of System 1, the Target Systems (and Components).
  - System 3, the System of Innovation, produces new abilities to do so, including knowledge.


4. Life Cycle Domains System
2. Target System (and Component) Life Cycle Domain System
3. System of Innovation (SOI)
1. Target System (and Components)

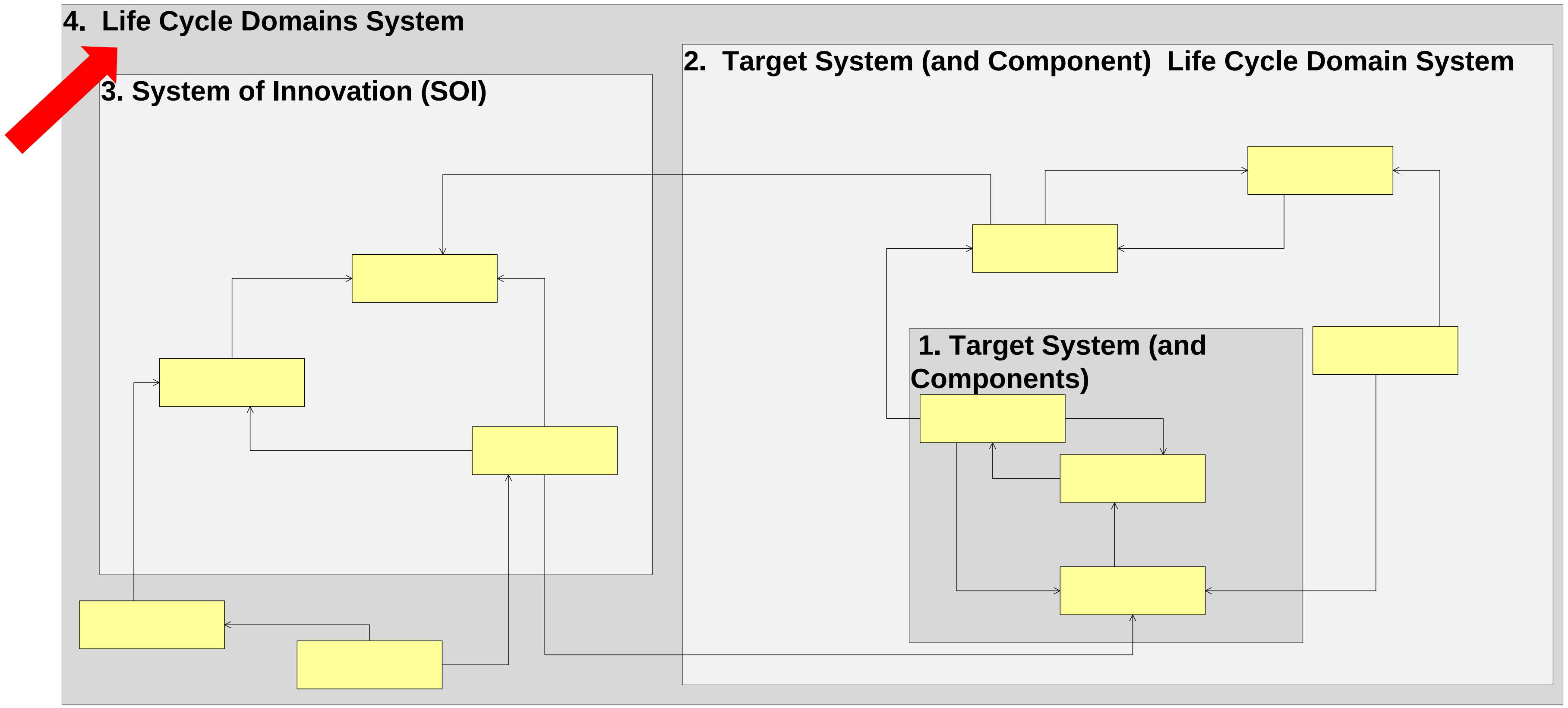
Example System of Innovation (System 3):
Home Entertainment System Example

- System Researchers, Designers: Electronic System Architects, Media and Device Basic Research Physical Scientists, Product & Process Designers, Network Architects, Computer Scientists, Standards Bodies.
- Configuration Management: CAD and PLM Tools and Information Systems.
- Security Management: Physical Security, Authentication, Authorization, Encryption


4. Life Cycle Domains System
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)

The Agile System Domain Model

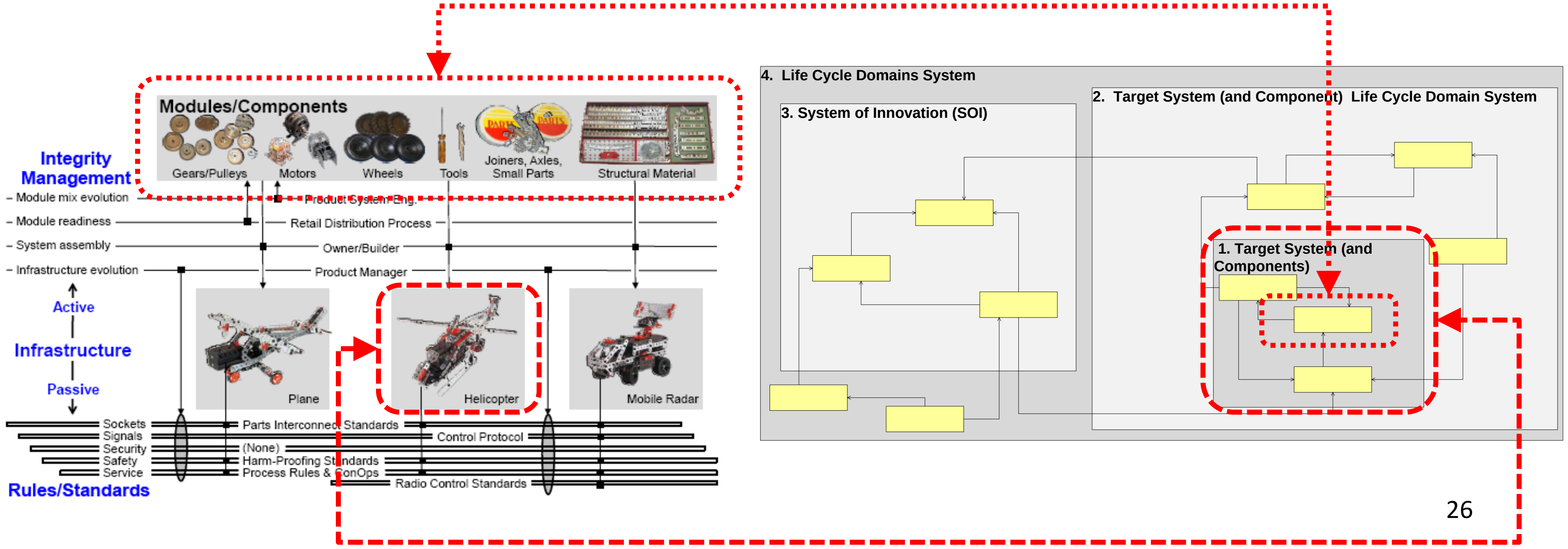
System 4: The **<u>Life Cycle Domains System</u>**, consisting of the entire environment of the Target System, along with that Target System, across all of its life cycle stages, including innovation:



4. Life Cycle Domains System
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)

So, up to this point, how much of the Agile System Architecture is visible?

- So far, probably most apparent is that:
  - The agile system can be identified with the Target System, and . . .
  - the agile system's "Modules Components" can be something similar to the Target System components, and . . .
  - differing aspects of the "Infrastructure" can be related to the Target System, the Target System Life Cycle Domain System, and the System of Innovation.
- However, this is still pretty vague, until we display more of the model . . .



Logical Architecture and Physical Architecture of the Target System

A. The Innovated (Target) System is partitioned into a collection of **<u>Target System Functional Roles</u>.** These interact with each other to create the externally visible "black box" behavior of the Target System:

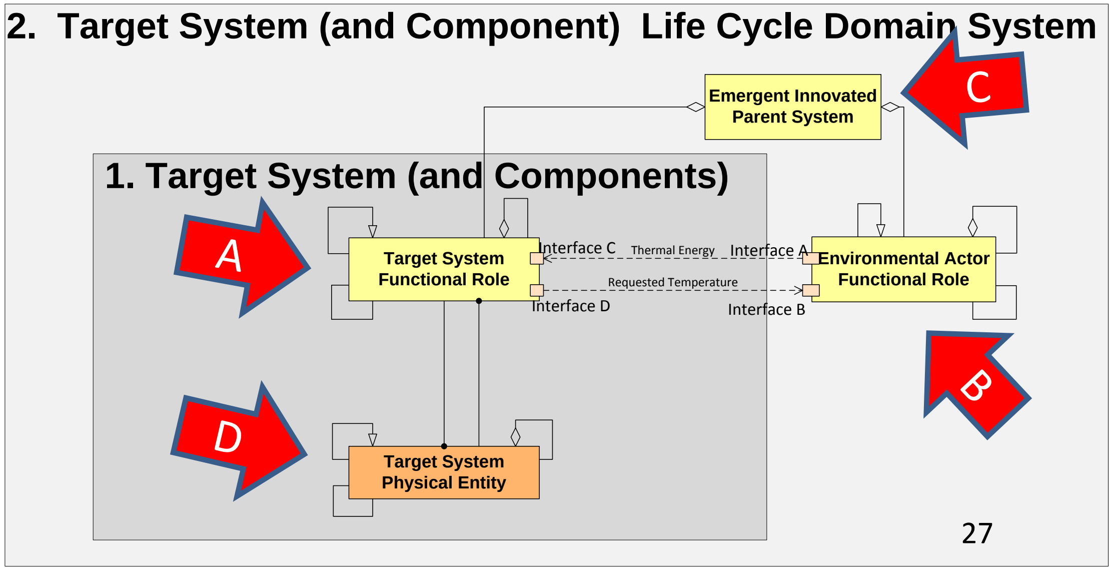
- The web of connected Functional Roles within that system is its Logical Architecture.
- These logical systems can also be in two types of hierarchy: A part-whole hierarchy and a special-general hierarchy.

B. The Innovated (Target) System interacts with external **<u>Environmental Actor Functional Roles</u>** played by environmental actors in the Target System (and Component) Life Cycle Domain System.

C. An **<u>Emergent Innovated Parent System</u>** is composed of the interacting Target System and its Environment.

D. The Target System Functional Roles are allocated to **<u>Target System Physical Entities</u>** that perform those roles:

- There can also be hierarchies of these.


2. Target System (and Component) Life Cycle Domain System
Emergent Innovated Parent System
C
1. Target System (and Components)
A
Target System Functional Role
Interface C
Thermal Energy
Interface A
Environmental Actor Functional Role
Requested Temperature
Interface D
Interface B
B
D
Target System Physical Entity

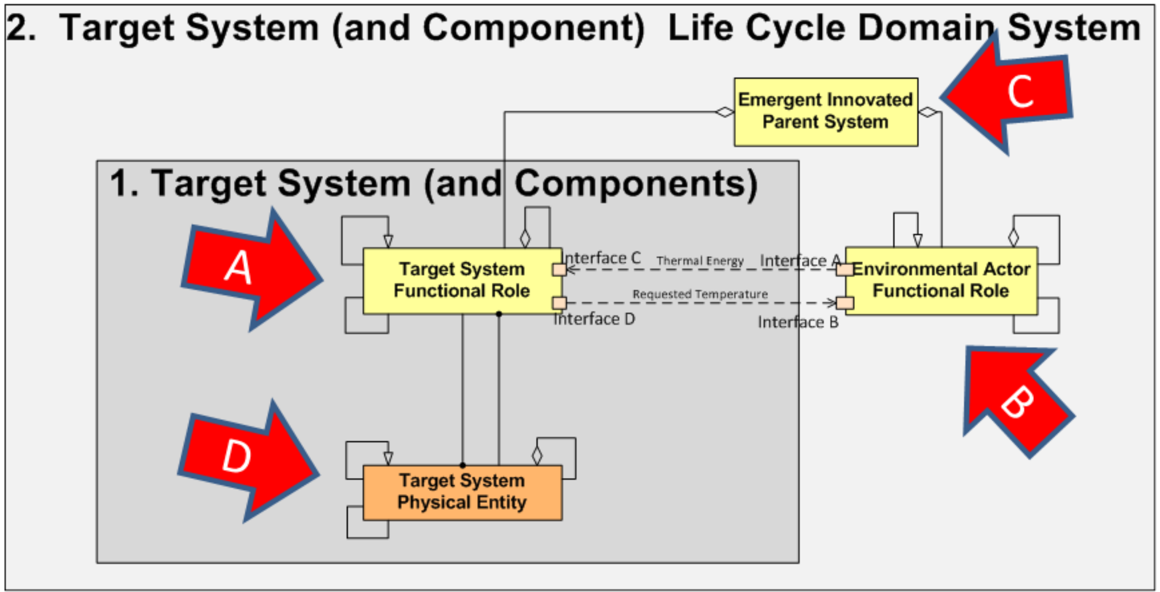
Example: Target System Roles, Environmental Actors, Physical Components, Emergent Innovated Parent System

A. Target System Functional Role: Downloadable Media Player

B. Environmental Actor Role: Music Library Supplier

C. Emergent Innovated Parent System: Post i-Tunes Music Industry

D. Physical Component: Apple iPhone6

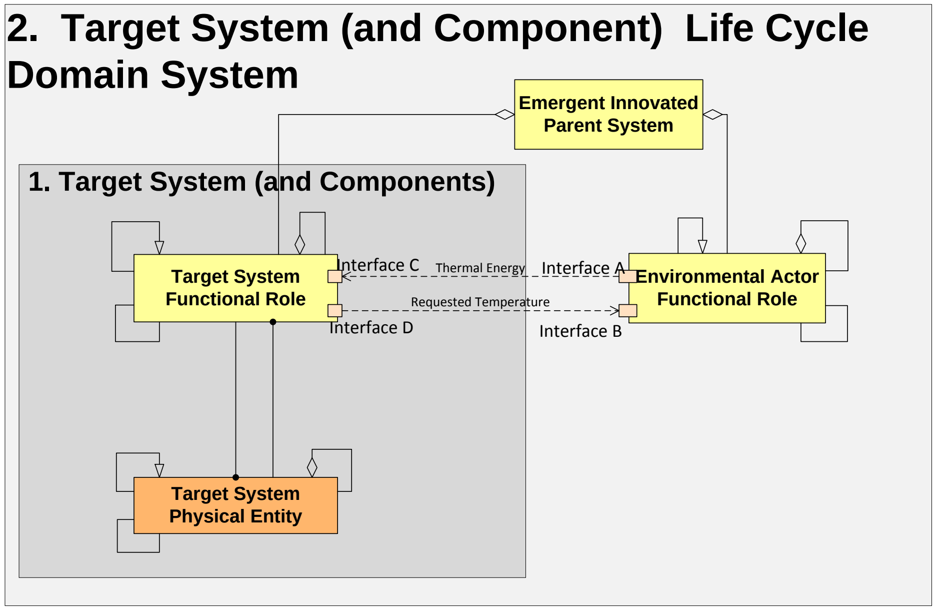

2. Target System (and Component) Life Cycle Domain System
Emergent Innovated Parent System
C
1. Target System (and Components)
A
Target System Functional Role
Interface C
Thermal Energy
Interface A
Environmental Actor Functional Role
Requested Temperature
Interface D
Interface B
B
D
Target System Physical Entity

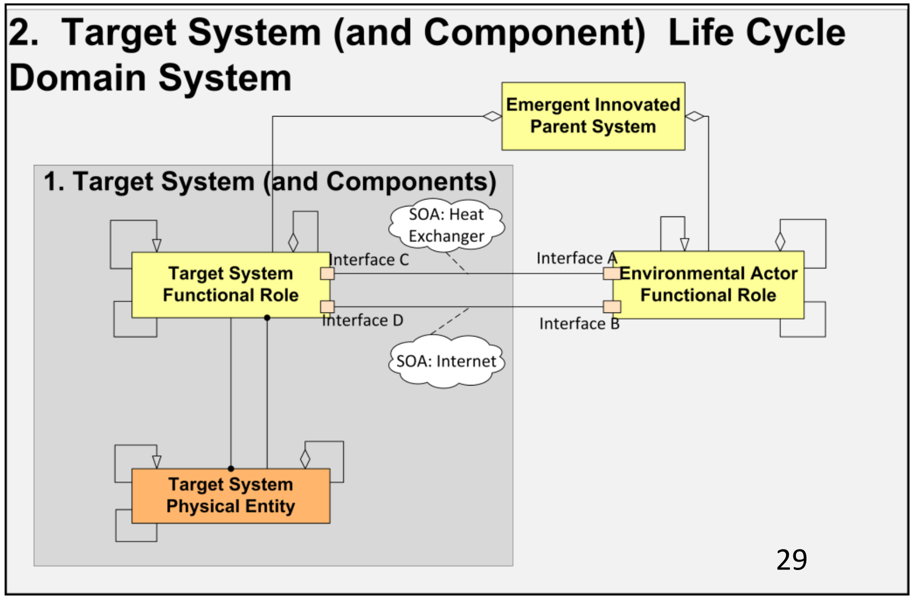
Logical Architecture and Physical Architecture of the Target System

A. Interacting roles exchange **Input-Outputs**, which are generally energies, forces, mass flows, or information.

B. These interactions occur through **Interfaces**, which associate:

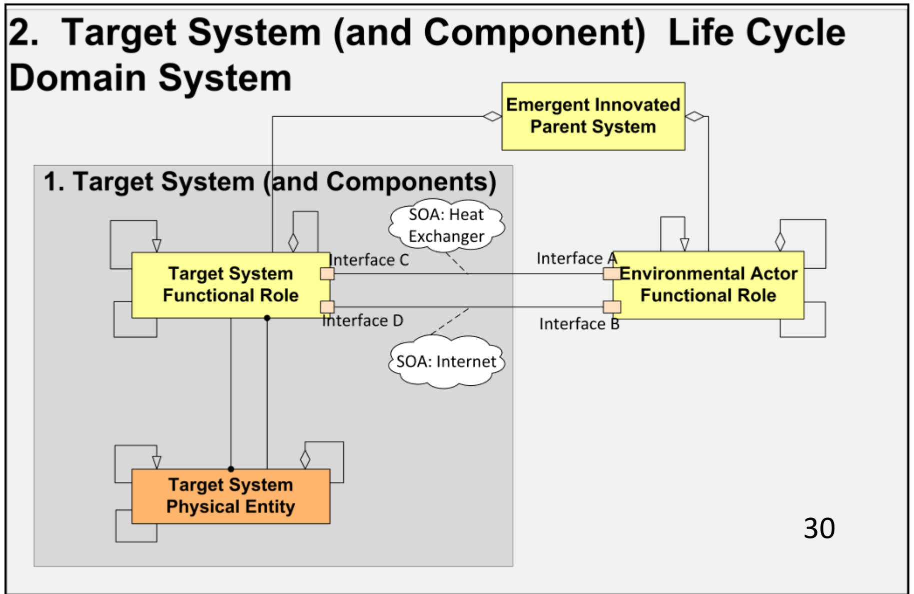
- Systems, that "have" the Interfaces
- Input-Outputs that pass through the Interface
- Interactions which describe behavior at the Interface
- **Systems of Access**, which are external intermediary "clouds" transporting the interaction Input-Output exchanges, between Interfaces


2. Target System (and Component) Life Cycle Domain System
Emergent Innovated Parent System
1. Target System (and Components)
Target System Functional Role
Interface C
Thermal Energy
Interface A
Environmental Actor Functional Role
Requested Temperature
Interface D
Interface B
Target System Physical Entity


2. Target System (and Component) Life Cycle Domain System
Emergent Innovated Parent System
1. Target System (and Components)
SOA: Heat Exchanger
Target System Functional Role
Interface C
Interface A
Environmental Actor Functional Role
Interface D
Interface B
SOA: Internet
Target System Physical Entity

Input-Outputs, Interfaces, Systems of Access: Home Entertainment System Example

- External Interfaces: Audio Output Interface; Visual Display Interface
- Input-Outputs: Music; Movies; Volume Control; Program Selection; Electrical Power; Heat; RFI; Vibration
- Systems of Access: Earbuds, Control Panels, Remote Controls, GUI


2. Target System (and Component) Life Cycle Domain System
Emergent Innovated Parent System
1. Target System (and Components)
SOA: Heat Exchanger
Interface C
Interface A
Target System Functional Role
Environmental Actor Functional Role
Interface D
Interface B
SOA: Internet
Target System Physical Entity

Logical Architecture and Physical Architecture of the Target System

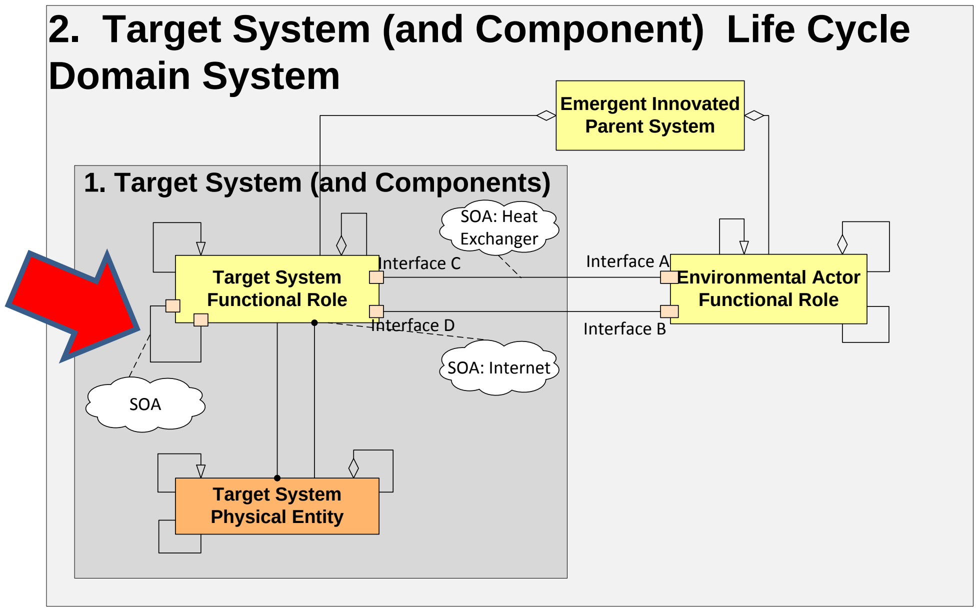
A. In like manner, there are interactions (exchanges of Input-Outputs) between roles within a Target System

B. There are internal Input-Outputs

C. There are internal Interfaces

D. There are internal Systems of Access

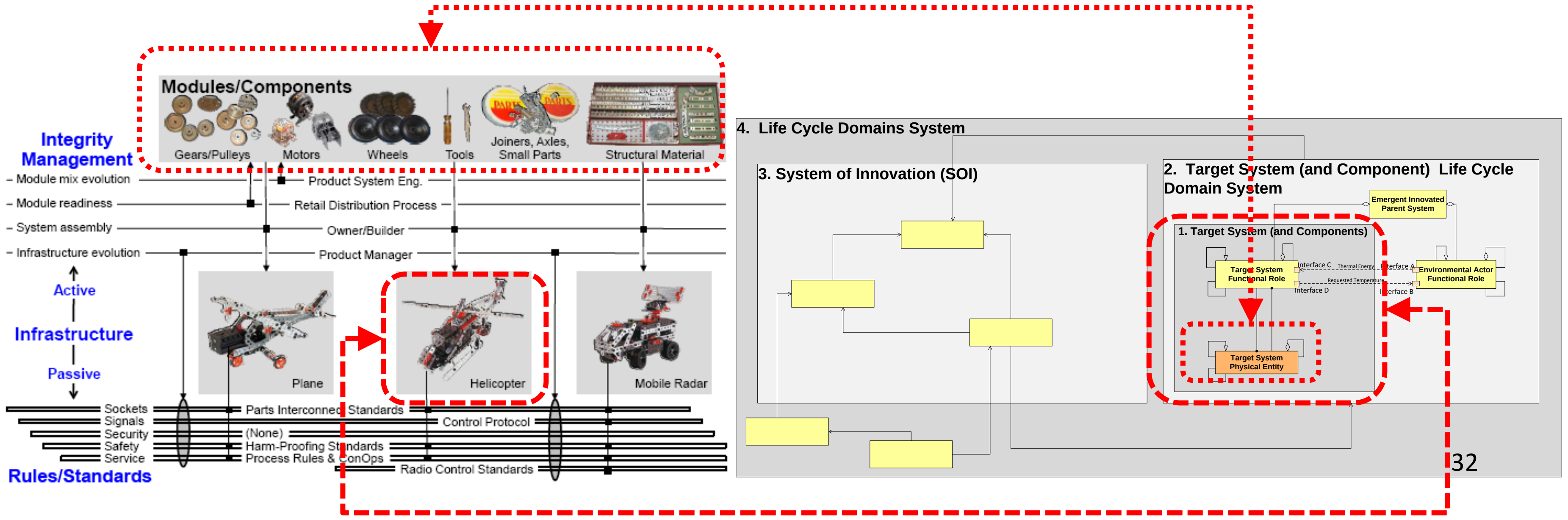
E. There are internal, composable relationships between these components



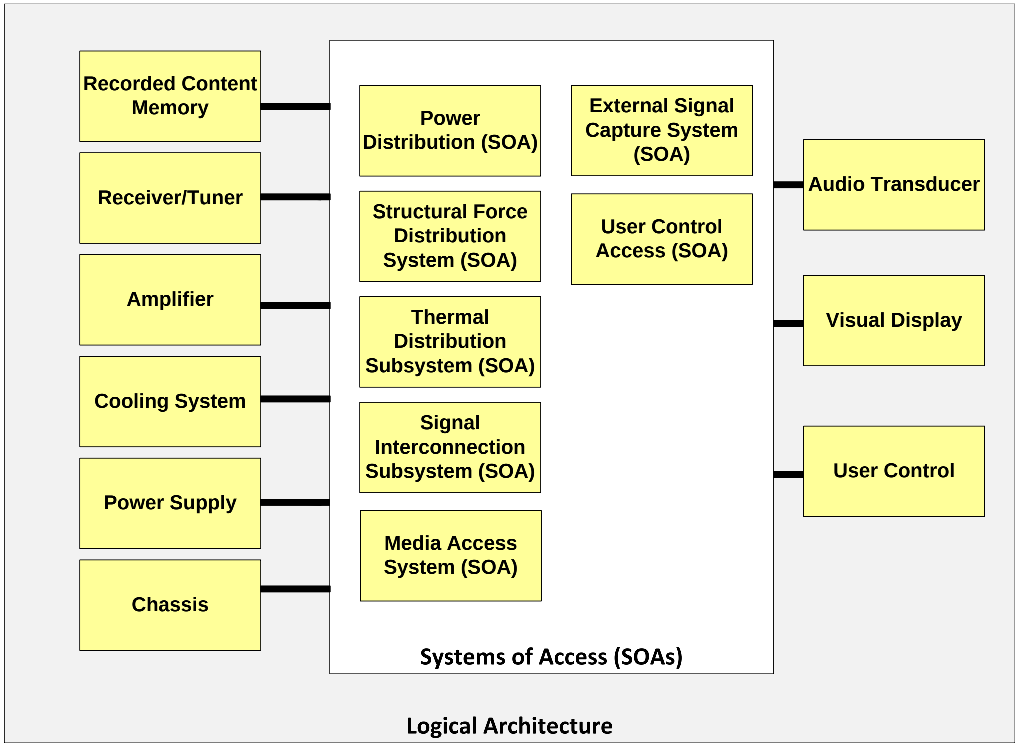
2. Target System (and Component) Life Cycle Domain System
Emergent Innovated Parent System
1. Target System (and Components)
SOA: Heat Exchanger
Interface C
Interface A
Target System Functional Role
Environmental Actor Functional Role
Interface D
Interface B
SOA: Internet
SOA
Target System Physical Entity

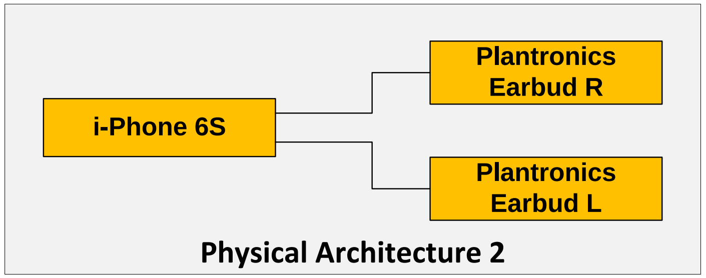
Logical Architecture and Physical Architecture of the Target System

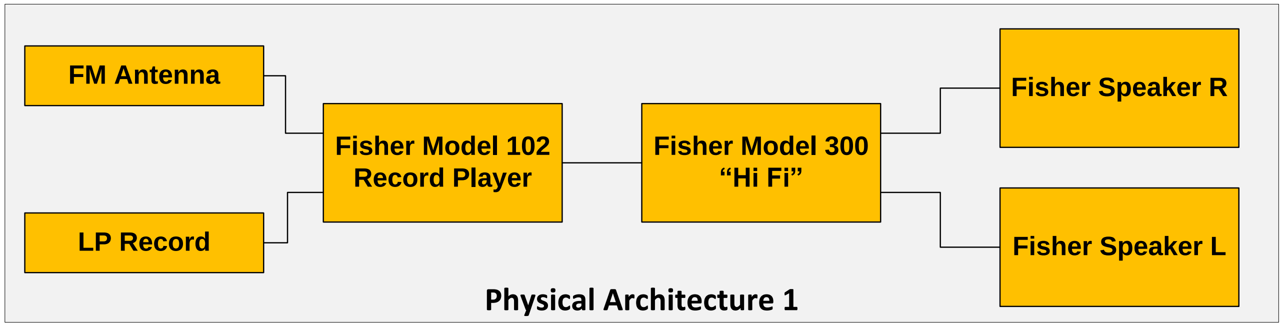
- So now we can see that the Agile System "Modules/Components":
  - are the Target System Physical Components,
  - which in turn play Target System Functional Roles,
  - Interacting with each other and external Environmental Actor Functional Roles.
- Remember that some of the Target System Physical Components:
  - may be "in inventory" for future integration into Target Systems.
  - may be humans, not just engineered parts--with related implications.



Example: Target System Logical and Physical Architecture, Internal Systems of Access


Recorded Content Memory
Receiver/Tuner
Amplifier
Cooling System
Power Supply
Chassis
Power Distribution (SOA)
Structural Force Distribution System (SOA)
Thermal Distribution Subsystem (SOA)
Signal Interconnection Subsystem (SOA)
Media Access System (SOA)
External Signal Capture System (SOA)
User Control Access (SOA)
Systems of Access (SOAs)
Audio Transducer
Visual Display
User Control
Logical Architecture



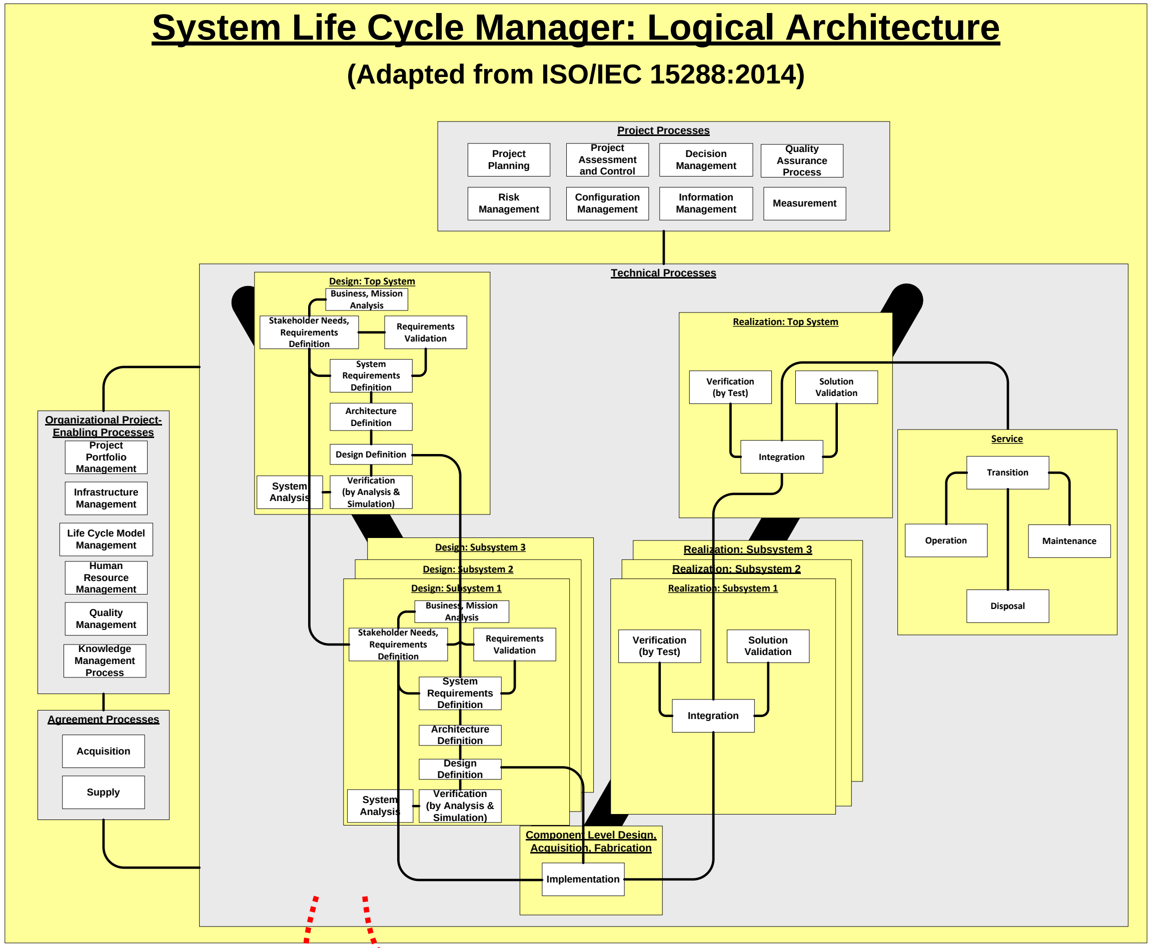


Relating Scrum and ISO 15288 Process Models

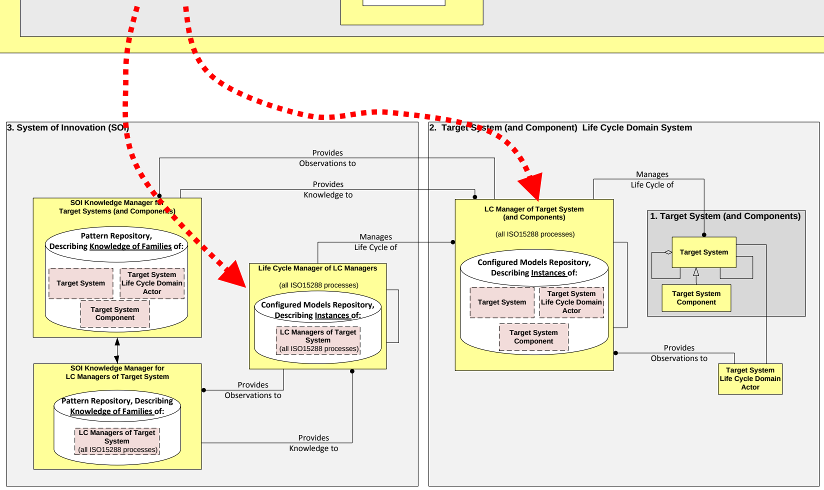
- More Than One Representation (Model View) of the Same Underlying Reality

(See Attachment I for more.)

ISO15288 Reference Processes



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

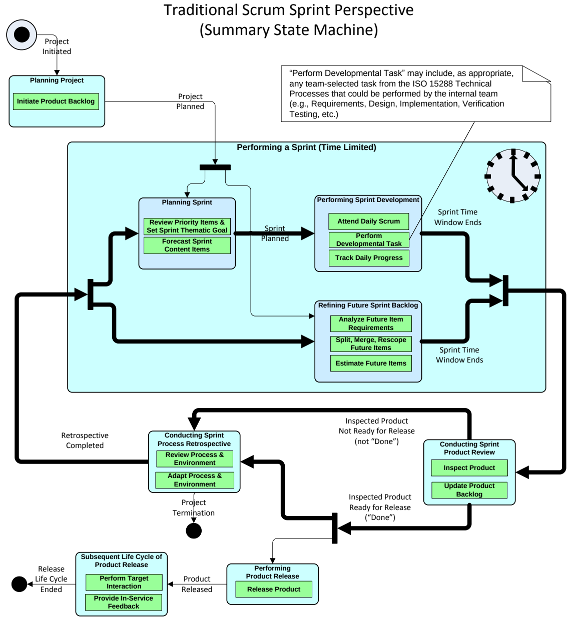


3. System of Innovation (SOI)
Provides Observations to
Provides Knowledge to
SOI Knowledge Manager for Target Systems (and Components)
Pattern Repository, Describing Knowledge of Families of:
Target System
Target System Life Cycle Domain Actor
Target System Component
Manages Life Cycle of
Life Cycle Manager of LC Managers
(all ISO15288 processes)
Configured Models Repository, Describing Instances of:
LC Managers of Target System (all ISO15288 processes)
SOI Knowledge Manager for LC Managers of Target System
Pattern Repository, Describing Knowledge of Families of:
LC Managers of Target System (all ISO15288 processes)
Provides Observations to
Provides Knowledge to
2. Target System (and Component) Life Cycle Domain System
Manages Life Cycle of
LC Manager of Target System (and Components)
(all ISO15288 processes)
Configured Models Repository, Describing Instances of:
Target System
Target System Life Cycle Domain Actor
Target System Component
1. Target System (and Components)
Target System
Target System Component
Provides Observations to
Target System Life Cycle Domain Actor

Agile Scrum Model

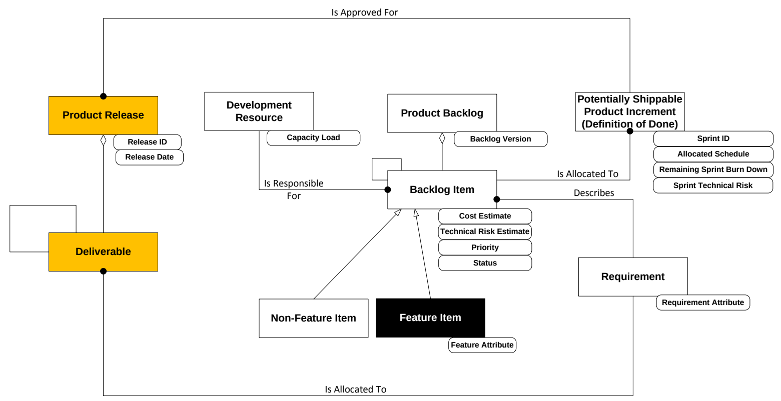
(See Attachment I for more.)

Traditional Scrum Sprint Perspective
(Summary State Machine)

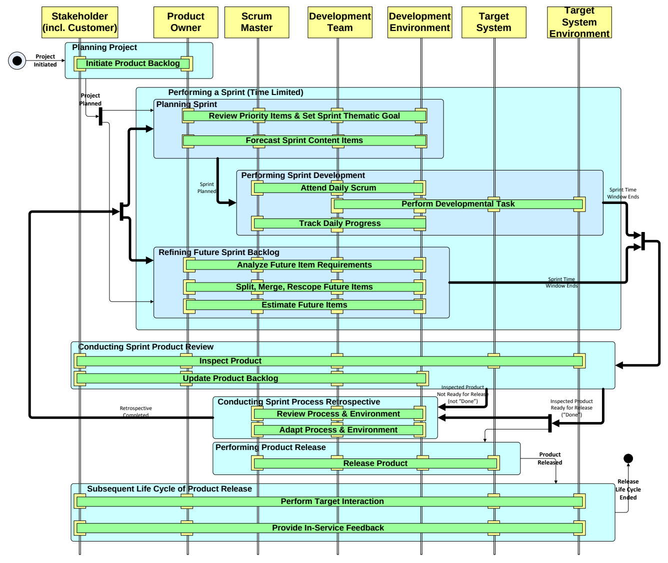


Project Initiated
Planning Project
Initiate Product Backlog
Project Planned
"Perform Developmental Task" may include, as appropriate, any team-selected task from the ISO 15288 Technical Processes that could be performed by the internal team (e.g., Requirements, Design, Implementation, Verification Testing, etc.)
Performing a Sprint (Time Limited)
Planning Sprint
Review Priority Items & Set Sprint Thematic Goal
Forecast Sprint Content Items
Sprint Planned
Performing Sprint Development
Attend Daily Scrum
Perform Developmental Task
Track Daily Progress
Sprint Time Window Ends
Refining Future Sprint Backlog
Analyze Future Item Requirements
Split, Merge, Rescope Future Items
Estimate Future Items
Sprint Time Window Ends
Retrospective Completed
Conducting Sprint Process Retrospective
Review Process & Environment
Adapt Process & Environment
Inspected Product Not Ready for Release (not "Done")
Conducting Sprint Product Review
Inspect Product
Update Product Backlog
Inspected Product Ready for Release ("Done")
Project Termination
Release Life Cycle Ended
Subsequent Life Cycle of Product Release
Perform Target Interaction
Provide In-Service Feedback
Product Released
Performing Product Release
Release Product

Traditional Scrum Sprint Perspective
(Simplified Model of Managed Information)

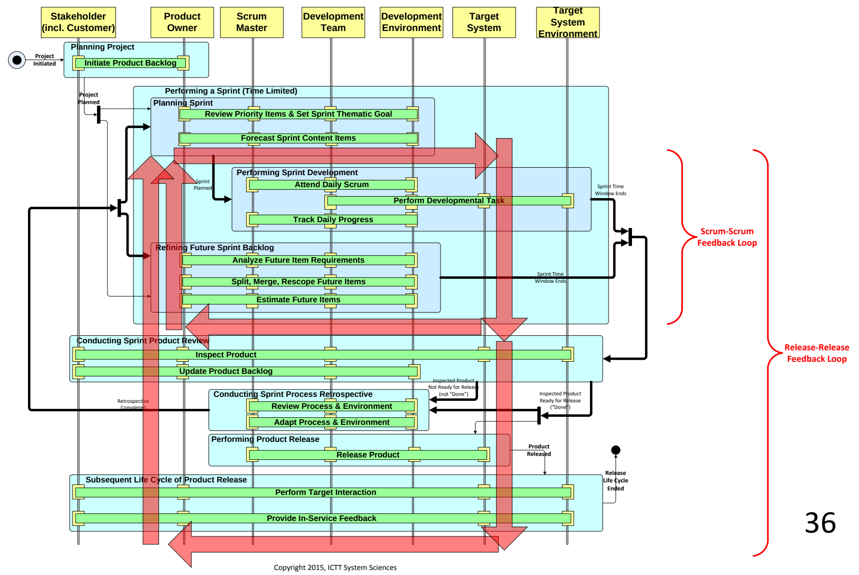


Traditional Scrum Sprint Perspective
(Activity Diagram, with Swim Lane Roles)



Stakeholder (incl. Customer)
Product Owner
Scrum Master
Development Team
Development Environment
Target System
Target System Environment
Project Initiated
Planning Project
Initiate Product Backlog
Project Planned
Performing a Sprint (Time Limited)
Planning Sprint
Review Priority Items & Set Sprint Thematic Goal
Forecast Sprint Content Items
Sprint Planned
Performing Sprint Development
Attend Daily Scrum
Perform Developmental Task
Track Daily Progress
Sprint Time Window Ends
Refining Future Sprint Backlog
Analyze Future Item Requirements
Split, Merge, Rescope Future Items
Estimate Future Items
Sprint Time Window Ends
Conducting Sprint Product Review
Inspect Product
Update Product Backlog
Inspected Product Not Ready for Release (not "Done")
Inspected Product Ready for Release ("Done")
Retrospective Completed
Conducting Sprint Process Retrospective
Review Process & Environment
Adapt Process & Environment
Performing Product Release
Release Product
Product Released
Release Life Cycle Ended
Subsequent Life Cycle of Product Release
Perform Target Interaction
Provide In-Service Feedback

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



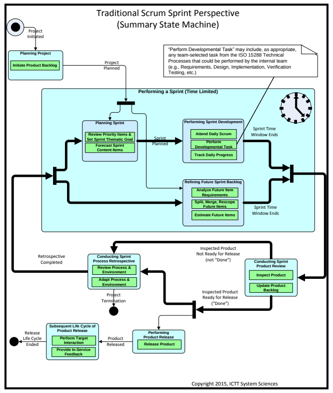
Stakeholder (incl. Customer)
Product Owner
Scrum Master
Development Team
Development Environment
Target System
Target System Environment
Project Initiated
Planning Project
Initiate Product Backlog
Project Planned
Performing a Sprint (Time Limited)
Planning Sprint
Review Priority Items & Set Sprint Thematic Goal
Forecast Sprint Content Items
Sprint Planned
Performing Sprint Development
Attend Daily Scrum
Perform Developmental Task
Track Daily Progress
Sprint Time Window Ends
Refining Future Sprint Backlog
Analyze Future Item Requirements
Split, Merge, Rescope Future Items
Estimate Future Items
Sprint Time Window Ends
Scrum-Scrum Feedback Loop
Release-Release Feedback Loop
Conducting Sprint Product Review
Inspect Product
Update Product Backlog
Inspected Product Not Ready for Release (not "Done")
Inspected Product Ready for Release ("Done")
Retrospective Completed
Conducting Sprint Process Retrospective
Review Process & Environment
Adapt Process & Environment
Performing Product Release
Release Product
Product Released
Release Life Cycle Ended
Subsequent Life Cycle of Product Release
Perform Target Interaction
Provide In-Service Feedback

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

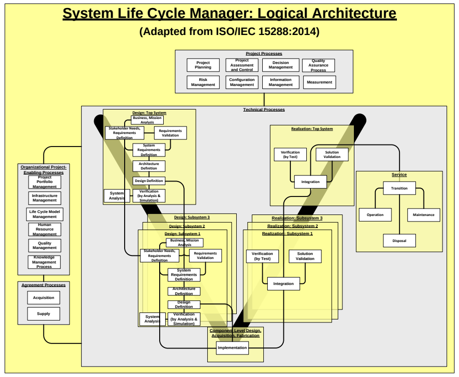
We are dealing with four different representations of the same underlying reality:

1. The Scrum Pattern: Emphasizes time and feedback, focusing on processes for learning 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



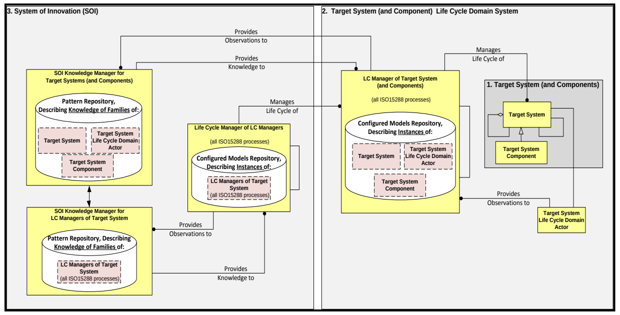
Traditional Scrum Sprint Perspective
(Summary State Machine)
Performing a Sprint (Time Limited)
Planning Sprint
Performing Sprint Development
Refining Future Sprint Backlog
Conducting Sprint Process Retrospective
Conducting Sprint Product Review
Performing Product Release
Copyright 2015, ICTT System Sciences



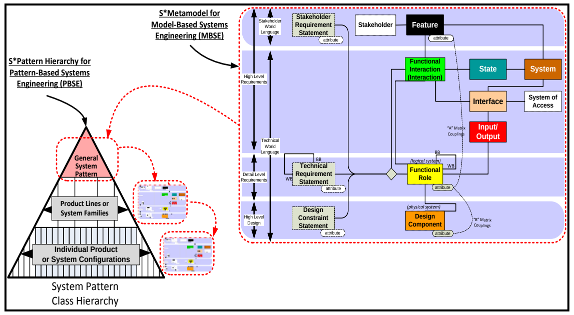
System Life Cycle Manager: Logical Architecture
(Adapted from ISO/IEC 15288:2014)
Project Processes
Technical Processes
Organizational Project-Enabling Processes
Agreement Processes
Service

ISO15288 Pattern

ASELC Pattern



3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)
SOI Knowledge Manager for Target Systems (and Components)
SOI Knowledge Manager for LC Managers of Target System
Life Cycle Manager of LC Managers
LC Manager of Target System (and Components)



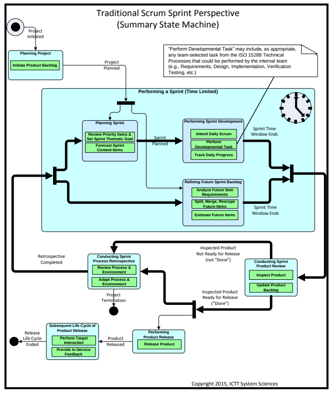
S*Metamodel for Model-Based Systems Engineering (MBSE)
S*Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
System Pattern Class Hierarchy
Stakeholder
Feature
Functional Interaction
State
System
Interface
Input/Output
Functional Role
Design Component

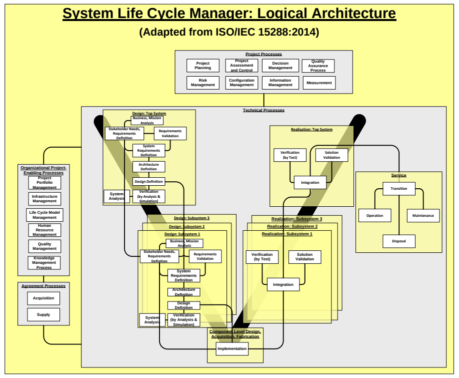
S*Metamodel

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

- The Scrum Model is actually an abstraction of the more complex-<u>looking</u> 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".

<u>Scrum Pattern</u>

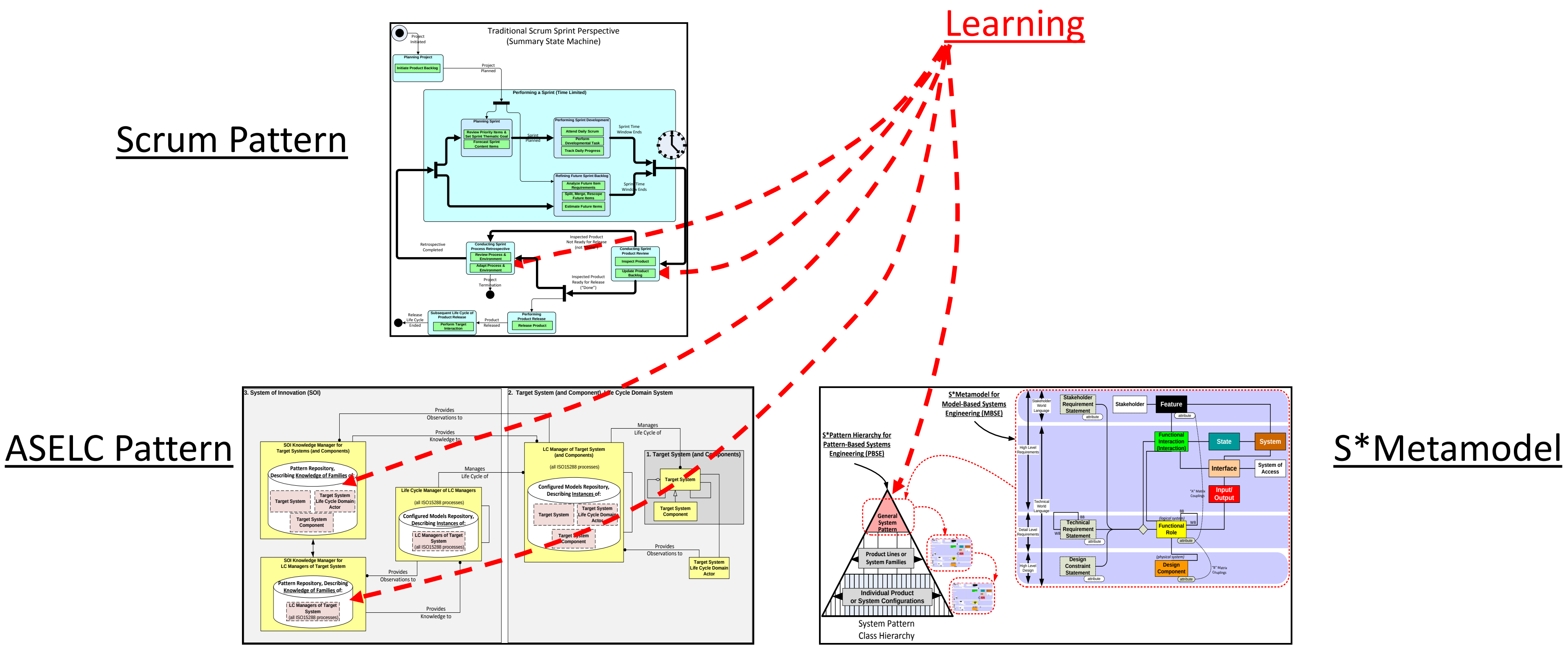

Traditional Scrum Sprint Perspective
(Summary State Machine)
Performing a Sprint (Time Limited)
Copyright 2015, ICTT System Sciences


System Life Cycle Manager: Logical Architecture
(Adapted from ISO/IEC 15288:2014)
Project Processes
Technical Processes
Organizational Project-Enabling Processes
Agreement Processes

<u>ISO15288 Pattern</u>

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

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


Learning
Scrum Pattern
Traditional Scrum Sprint Perspective (Summary State Machine)
ASELC Pattern
3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
S*Metamodel
S*Metamodel for Model-Based Systems Engineering (MBSE)
S*Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
System Pattern Class Hierarchy

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

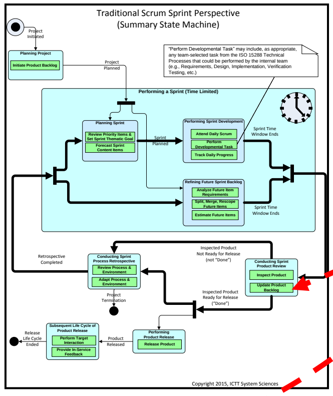
More Than One Representation (Model View) of the Same Underlying Reality

- Notice that the division of the System 3 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)

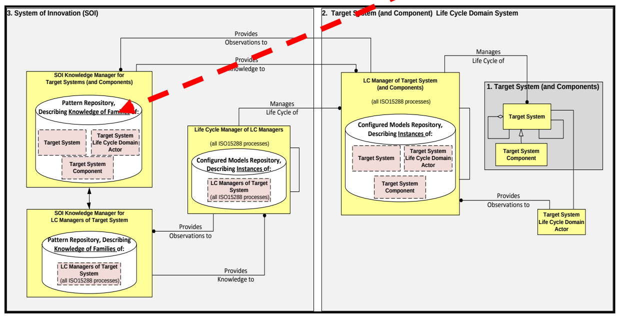
Pattern: Learnings about Development / Fielding System & Its Environment

Scrum Pattern

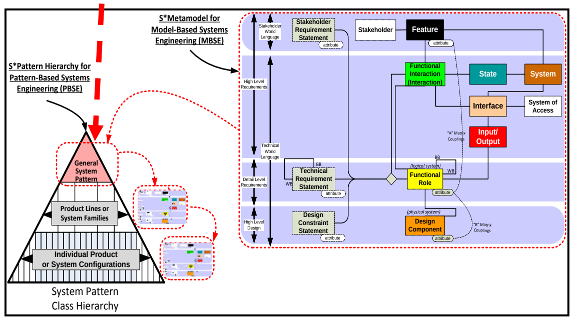


Traditional Scrum Sprint Perspective
(Summary State Machine)
Project Initiated
Planning Project
Initiate Product Backlog
Project Planned
"Perform Developmental Task" may include, as appropriate, any team-selected task from the ISO 15288 Technical Processes that could be performed by the internal team (e.g., Requirements, Design, Implementation, Verification Testing, etc.)
Performing a Sprint (Time Limited)
Planning Sprint
Review Priority Items & Set Sprint Thematic Goal
Forecast Sprint Content Items
Sprint Planned
Performing Sprint Development
Attend Daily Scrum
Perform Developmental Task
Track Daily Progress
Sprint Time Window Ends
Refining Future Sprint Backlog
Analyze Future Item Requirements
Split, Merge, Rescope Future Items
Estimate Future Items
Sprint Time Window Ends
Retrospective Completed
Conducting Sprint Process Retrospective
Review Process & Environment
Adapt Process & Environment
Inspected Product Not Ready for Release (not "Done")
Conducting Sprint Product Review
Inspect Product
Update Product Backlog
Project Termination
Inspected Product Ready for Release ("Done")
Release Life Cycle Ended
Subsequent Life Cycle of Product Release
Perform Target Interaction
Provide In-Service Feedback
Product Released
Performing Product Release
Release Product
Copyright 2015, ICTT System Sciences

ASELC Pattern



3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
Provides Observations to
Provides Knowledge to
SOI Knowledge Manager for Target Systems (and Components)
Pattern Repository, Describing Knowledge of Families of:
Target System
Target System Life Cycle Domain Actor
Target System Component
Manages Life Cycle of
Life Cycle Manager of LC Managers
(all ISO15288 processes)
Configured Models Repository, Describing Instances of:
LC Managers of Target System
(all ISO15288 processes)
LC Manager of Target System (and Components)
(all ISO15288 processes)
Configured Models Repository, Describing Instances of:
Target System
Target System Life Cycle Domain Actor
Target System Component
Manages Life Cycle of
1. Target System (and Components)
Target System
Target System Component
Provides Observations to
Target System Life Cycle Domain Actor
SOI Knowledge Manager for LC Managers of Target System
Pattern Repository, Describing Knowledge of Families of:
LC Managers of Target System
(all ISO15288 processes)
Provides Observations to
Provides Knowledge to



S*Metamodel for Model-Based Systems Engineering (MBSE)
S*Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
System Pattern Class Hierarchy
Stakeholder World Language
Stakeholder Requirement Statement
Stakeholder
Feature
High Level Requirements
Functional Interaction (Interaction)
State
System
Interface
System of Access
Input/ Output
Technical World Language
Technical Requirement Statement
Detail Level Requirements
Functional Role
High Level Design
Design Constraint Statement
Design Component
attribute

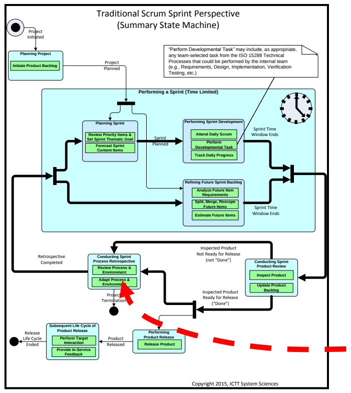
S*Metamodel

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

- Notice that the division of the System 3 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)

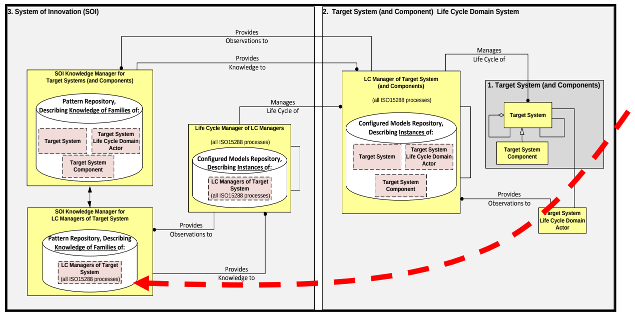
Scrum Pattern



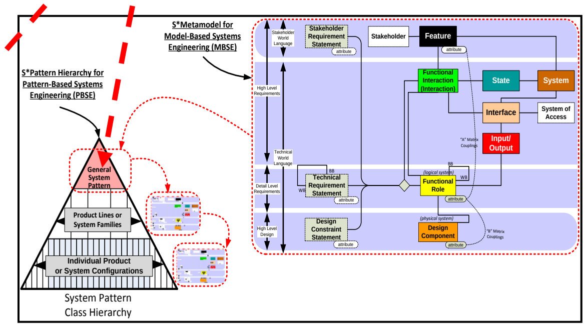
Traditional Scrum Sprint Perspective
(Summary State Machine)
Planning Project
Initiate Product Backlog
Project Planned
Performing a Sprint (Time Limited)
Planning Sprint
Performing Sprint Development
Attend Daily Scrum
Perform Developmental Task
Track Daily Progress
Refining Future Sprint Backlog
Sprint Time Window Ends
Conducting Sprint Process Retrospective
Review Process & Environment
Adapt Process & Environment
Retrospective Completed
Conducting Sprint Product Review
Inspect Product
Update Product Backlog
Inspected Product Not Ready for Release (not "Done")
Inspected Product Ready for Release ("Done")
Project Termination
Subsequent Life Cycle of Product Release
Perform Target Interaction
Provide In-Service Feedback
Performing Product Release
Release Product
Product Released
Release Life Cycle Ended
Copyright 2015, ICTT System Sciences

Pattern: Learnings about Development / Fielding System & Its Environment

ASELC Pattern



3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
1. Target System (and Components)
Provides Observations to
Provides Knowledge to
Manages Life Cycle of
SOI Knowledge Manager for Target Systems (and Components)
Pattern Repository, Describing Knowledge of Families of:
Target System
Target System Life Cycle Domain Actor
Target System Component
Life Cycle Manager of LC Managers
Configured Models Repository, Describing Instances of:
LC Managers of Target System
LC Manager of Target System (and Components)
SOI Knowledge Manager for LC Managers of Target System
Target System Life Cycle Domain Actor

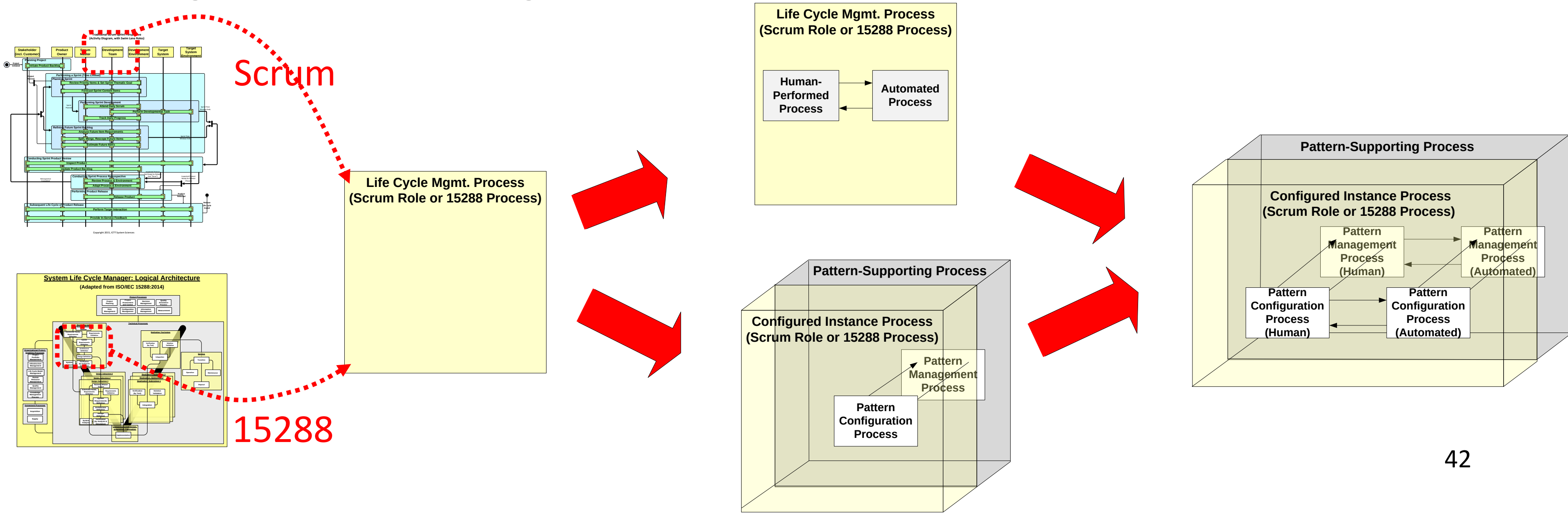


S*Metamodel for Model-Based Systems Engineering (MBSE)
S*Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
System Pattern Class Hierarchy
Stakeholder Requirement Statement
Stakeholder
Feature
Functional Interaction (Interaction)
State
System
Interface
System of Access
Input/Output
Technical Requirement Statement
Functional Role
Design Constraint Statement
Design Component

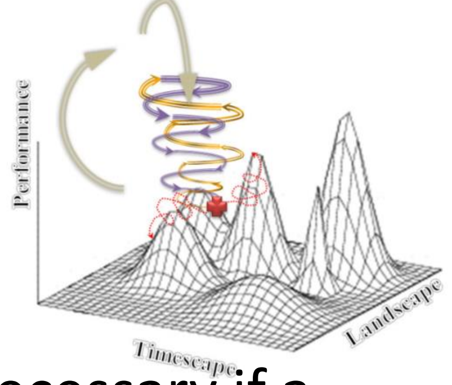
S*Metamodel

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

- Although the Scrum Model purposely simplifies those complexities, for ease of understanding "agility", a complete analysis of the Agile SE Process requires that the process analyst understand their formal representation in the context of the Agile SE Process Model.
- This includes how each of the Scrum Model and 15288 Model "roles" can be decomposed two different ways:
  - (Human Agent) interacting with (Automated Information System Agent)
  - (Management of Current System Configuration) interacting with (Management of Pattern)
- Resulting in four interacting internal roles:

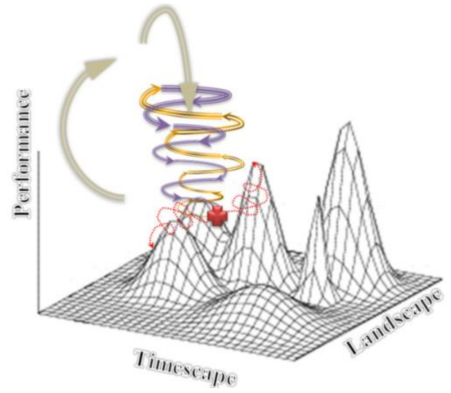

Scrum
System Life Cycle Manager: Logical Architecture
(Adapted from ISO/IEC 15288:2014)
15288
Life Cycle Mgmt. Process
(Scrum Role or 15288 Process)
Life Cycle Mgmt. Process
(Scrum Role or 15288 Process)
Human-Performed Process
Automated Process
Pattern-Supporting Process
Configured Instance Process
(Scrum Role or 15288 Process)
Pattern Management Process
Pattern Configuration Process
Pattern-Supporting Process
Configured Instance Process
(Scrum Role or 15288 Process)
Pattern Management Process (Human)
Pattern Management Process (Automated)
Pattern Configuration Process (Human)
Pattern Configuration Process (Automated)

Necessary & Sufficient for Representing Agility



- Necessary:
  - We assert that all the logical systems listed are necessary if a Target System is to avoid being “out agiled” by another Target System.
  - By this, we mean having a more agile response trajectory in configuration space.
- To prove this:
  - Suppose a candidate Agile System omits any one of the logical subsystems described (Systems 1-3).
  - Then we can describe a situation in which a competing Agile System candidate will out-perform it in the response sense.
- This does not mean that the decomposition is unique--only that what it covers cannot be omitted across all configurations of agility.

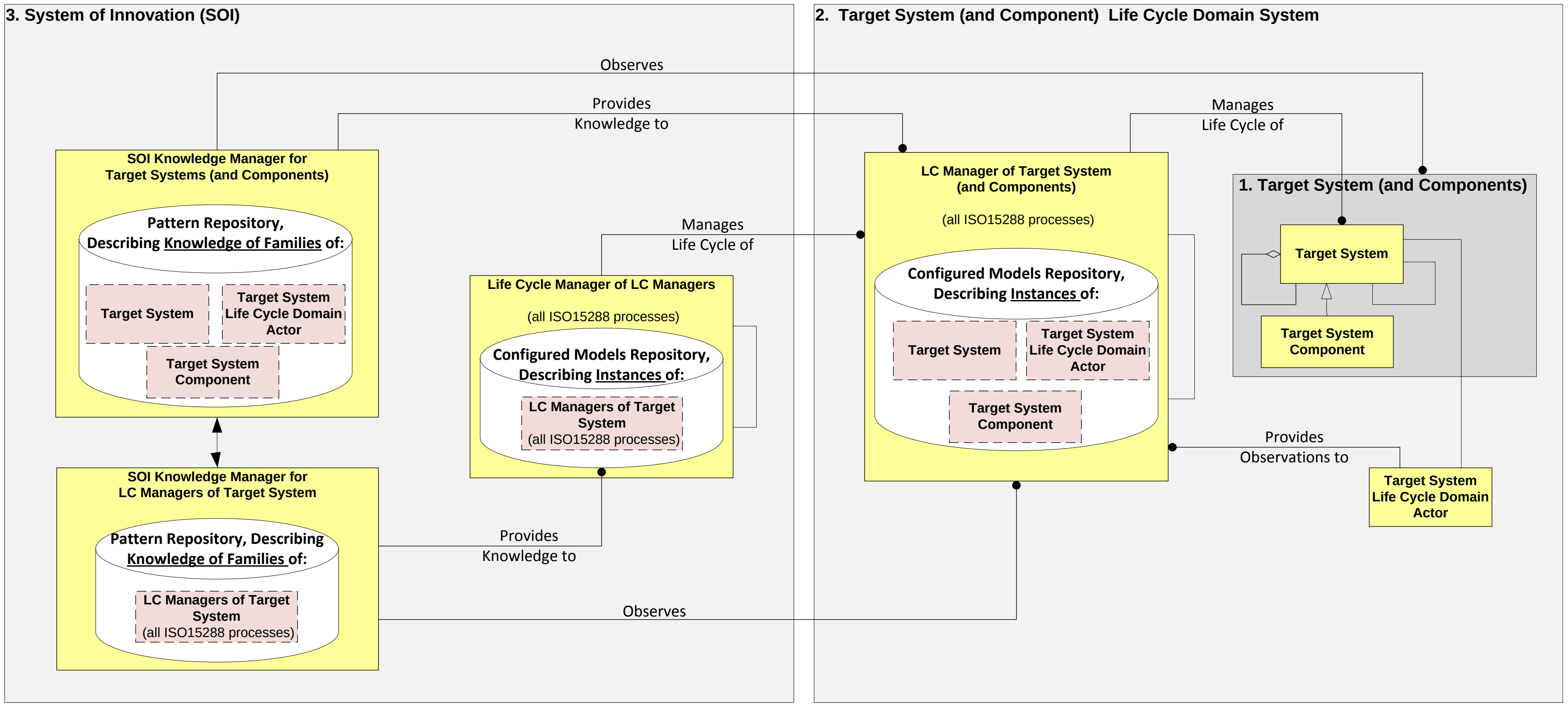
Necessary & Sufficient for Representing Agility



- Sufficient:
  - The fact that a candidate Agile System conforms to this pattern does not mean that it is more agile than a competing Agile System:
    - Two candidate systems conforming to this framework can exhibit different levels of agility, . . .
    - based on the pattern content accumulated by their System 3s, used to drive their Systems 1 and 2.
  - However, the framework is sufficient to describe any such Agile System competitors:
    - In fact, it can also describe, for comparison, very “un-agile” competitors, along with highly agile cases.
  - This is why it is a descriptive framework for use in the ASELCM Project

Agile System Features Model

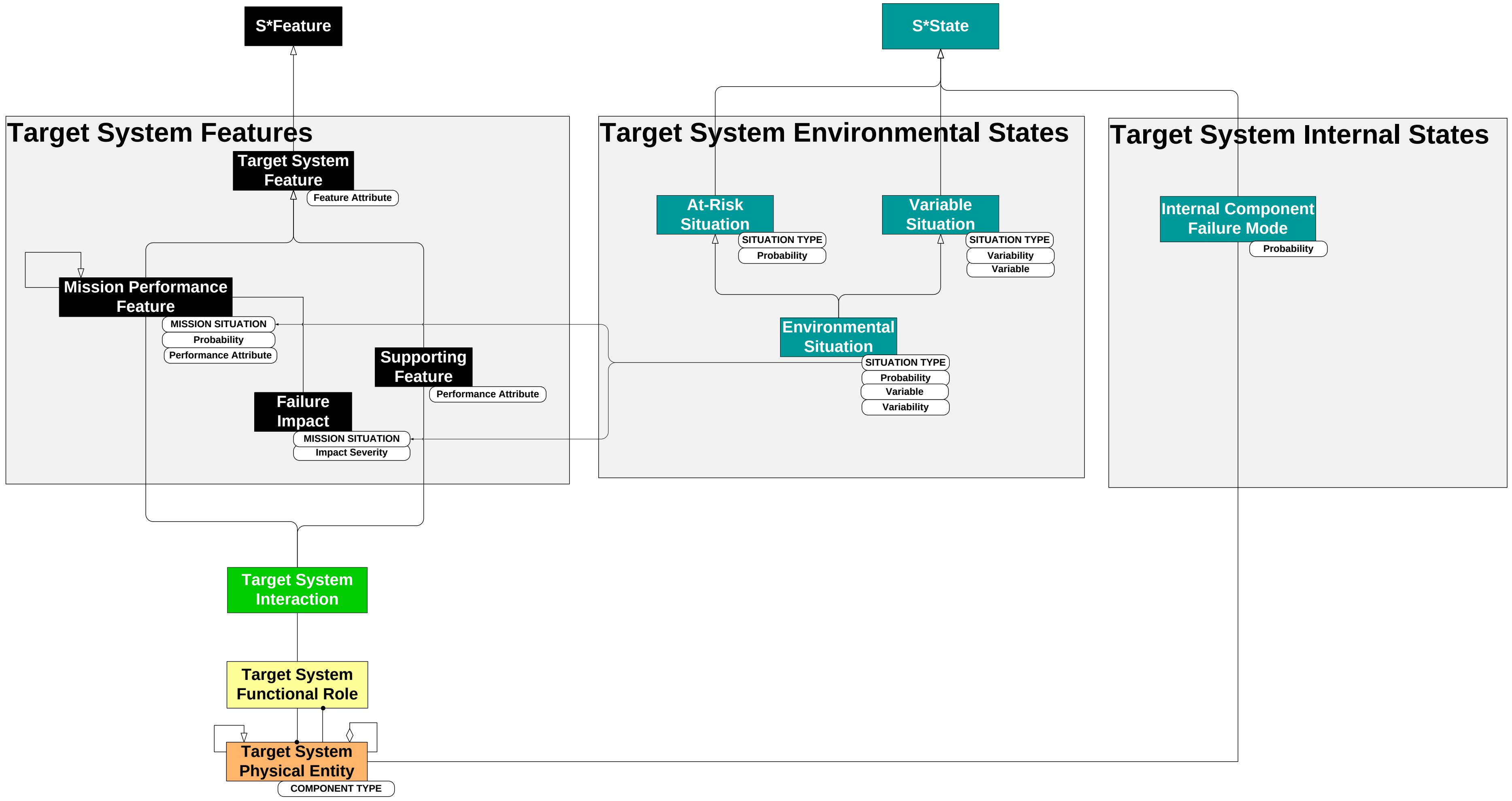
- Various systems have different features
- Reminder of Systems 1, 2, and 3--


3. System of Innovation (SOI)
2. Target System (and Component) Life Cycle Domain System
Observes
Provides Knowledge to
Manages Life Cycle of
SOI Knowledge Manager for Target Systems (and Components)
Pattern Repository, Describing Knowledge of Families of:
Target System
Target System Life Cycle Domain Actor
Target System Component
LC Manager of Target System (and Components)
(all ISO15288 processes)
1. Target System (and Components)
Target System
Target System Component
Manages Life Cycle of
Life Cycle Manager of LC Managers
(all ISO15288 processes)
Configured Models Repository, Describing Instances of:
LC Managers of Target System (all ISO15288 processes)
Configured Models Repository, Describing Instances of:
Target System
Target System Life Cycle Domain Actor
Target System Component
Provides Observations to
Target System Life Cycle Domain Actor
SOI Knowledge Manager for LC Managers of Target System
Pattern Repository, Describing Knowledge of Families of:
LC Managers of Target System (all ISO15288 processes)
Provides Knowledge to
Observes

Agile System Pattern: Major Feature Groups

- **System 1 Features**: Stakeholder capabilities of the Target System—the system we ultimately want to respond (with help from Systems 2 and 3) in agile fashion:
  - Example: Amplifier Multimedia Compatibility Feature
- **System 2 Features**: Stakeholder capabilities of the Target System Life Cycle Management System. This includes all aspects of its LC, a subset of which are relevant to the Agile Systems LC Pattern.
  - Example: Audio System Component Retailer Delivery Time Feature
- **System 3 Features**: Stakeholder capabilities of the three subsystems of System 3—concerned with observing and learning about the Target System and its Environment, and about the Target System LC Manager; also responsible for managing the LC of the Target System LC Manager.
  - Example: Audio Manufacturer Emergent Standards Compliance Feature

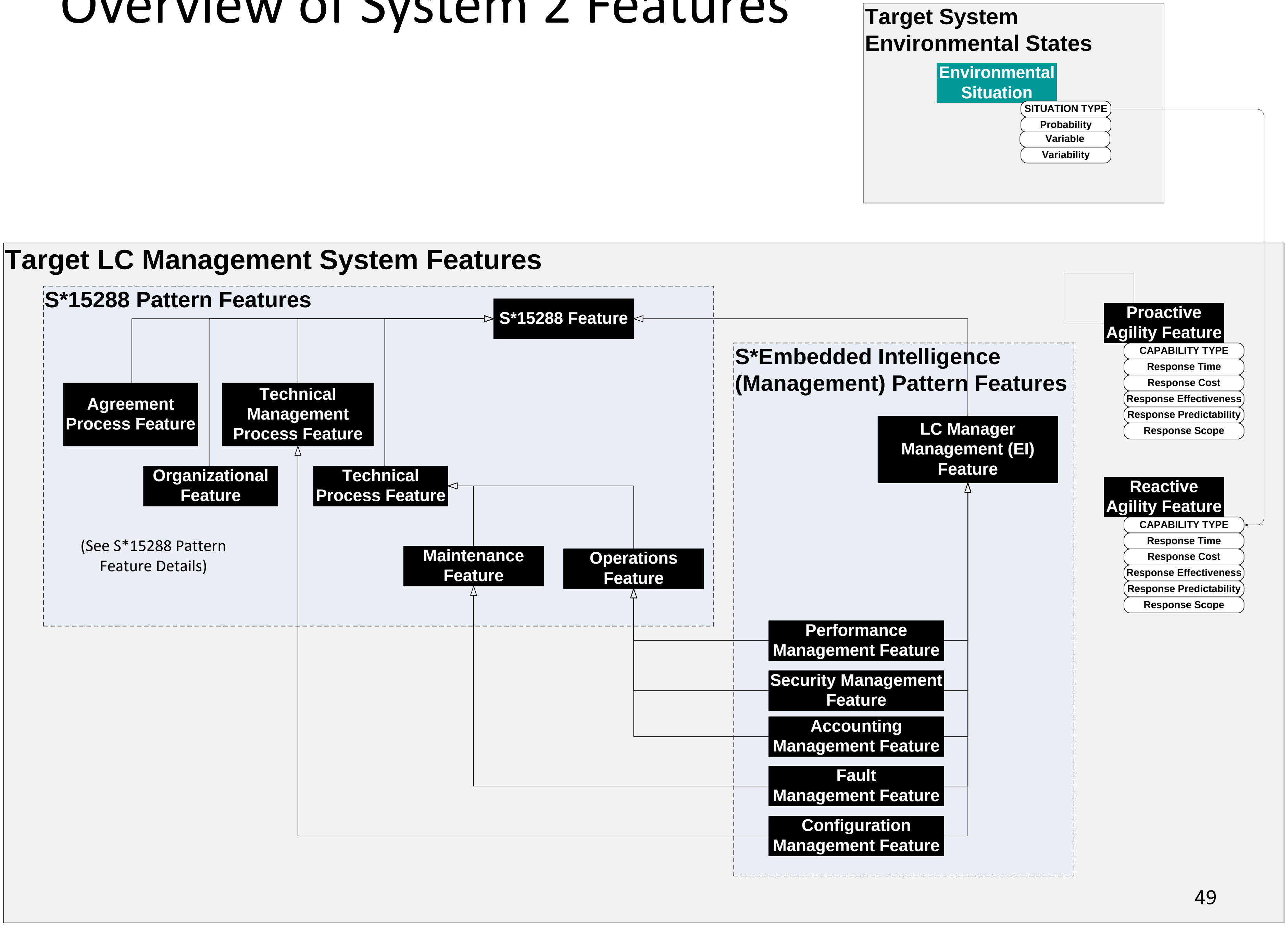
Overview of System 1 Features (and some related S*Metaclasses)



Subset of Features: Life Cycle Management Features

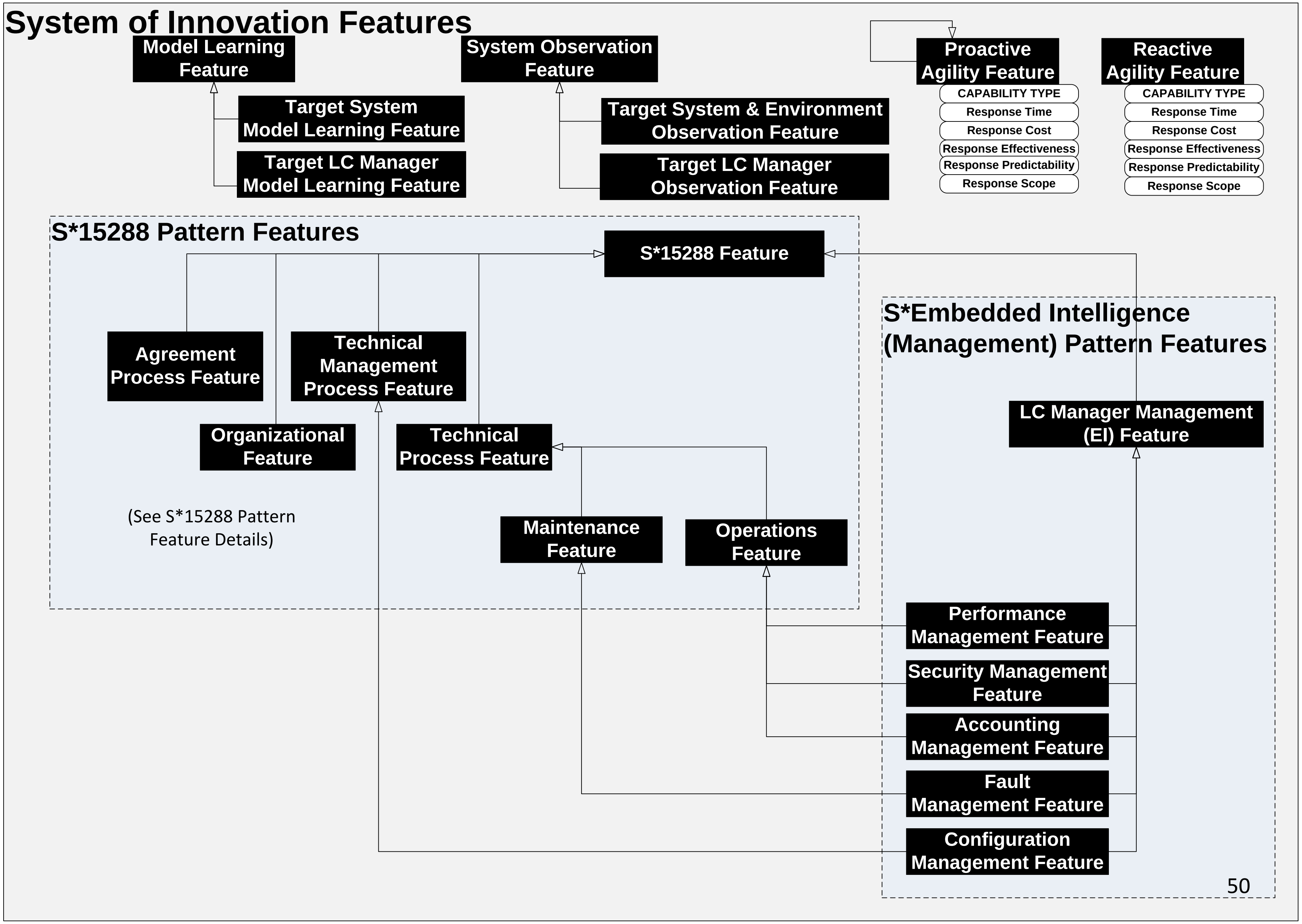
- These represent the stakeholder level view of capabilities to manage life cycles of either System 1 or System 2.
- We are making use of the S*15288 Pattern, which models ISO 15288 process capabilities:
  - This pattern is a specialization of the more abstract features of the generic Systems of Innovation (SOI) Pattern (Beihoff and Schindel, 2012)

Overview of System 2 Features



Target System Environmental States
Environmental Situation
SITUATION TYPE
Probability
Variable
Variability
Target LC Management System Features
S*15288 Pattern Features
S*15288 Feature
Agreement Process Feature
Technical Management Process Feature
Organizational Feature
Technical Process Feature
(See S*15288 Pattern Feature Details)
Maintenance Feature
Operations Feature
S*Embedded Intelligence (Management) Pattern Features
LC Manager Management (EI) Feature
Performance Management Feature
Security Management Feature
Accounting Management Feature
Fault Management Feature
Configuration Management Feature
Proactive Agility Feature
CAPABILITY TYPE
Response Time
Response Cost
Response Effectiveness
Response Predictability
Response Scope
Reactive Agility Feature
CAPABILITY TYPE
Response Time
Response Cost
Response Effectiveness
Response Predictability
Response Scope

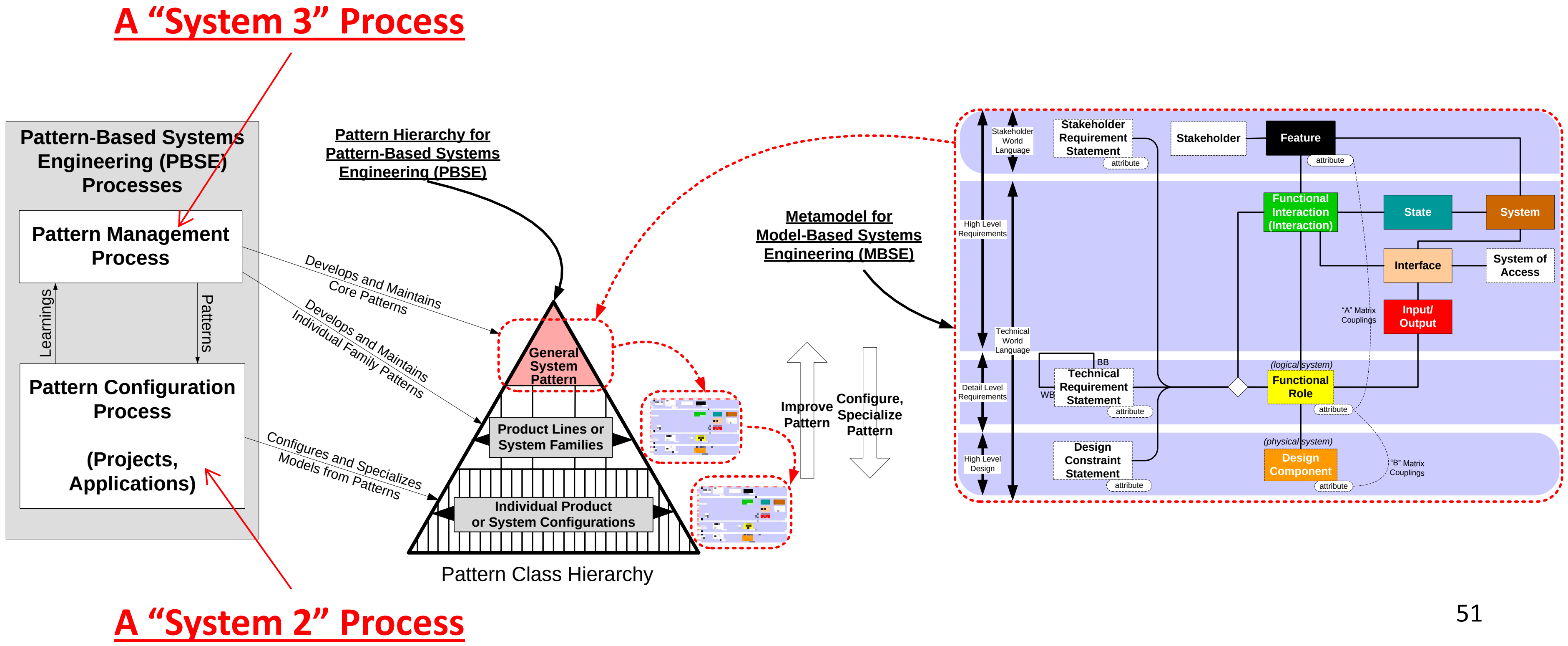
Overview of System 3 Features



System of Innovation Features
Model Learning Feature
Target System Model Learning Feature
Target LC Manager Model Learning Feature
System Observation Feature
Target System & Environment Observation Feature
Target LC Manager Observation Feature
Proactive Agility Feature
CAPABILITY TYPE
Response Time
Response Cost
Response Effectiveness
Response Predictability
Response Scope
Reactive Agility Feature
CAPABILITY TYPE
Response Time
Response Cost
Response Effectiveness
Response Predictability
Response Scope
S*15288 Pattern Features
S*15288 Feature
Agreement Process Feature
Technical Management Process Feature
Organizational Feature
Technical Process Feature
(See S*15288 Pattern Feature Details)
Maintenance Feature
Operations Feature
S*Embedded Intelligence (Management) Pattern Features
LC Manager Management (EI) Feature
Performance Management Feature
Security Management Feature
Accounting Management Feature
Fault Management Feature
Configuration Management Feature

Practical Implications for Agile Modeling

- In fewest words:
  - PBSE is agile MBSE.
  - (System 2 people should) Learn the Model, not Modeling.
- A small number of System 3 people can make a large number of System 2 people more agile.


A “System 3” Process
Pattern-Based Systems Engineering (PBSE) Processes
Pattern Management Process
Learnings
Patterns
Pattern Configuration Process
(Projects, Applications)
A “System 2” Process
Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
Develops and Maintains Core Patterns
Develops and Maintains Individual Family Patterns
Configures and Specializes Models from Patterns
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
Pattern Class Hierarchy
Improve Pattern
Configure, Specialize Pattern
Metamodel for Model-Based Systems Engineering (MBSE)
Stakeholder World Language
Stakeholder Requirement Statement
attribute
Stakeholder
Feature
High Level Requirements
Functional Interaction (Interaction)
State
System
Interface
System of Access
"A" Matrix Couplings
Input/ Output
Technical World Language
Detail Level Requirements
BB
WB
Technical Requirement Statement
(logical system)
Functional Role
High Level Design
Design Constraint Statement
(physical system)
Design Component
"B" Matrix Couplings

System Patterns Answer a Key Challenge to Agile Methods Adopters

- Another hallmark of agile methods is the repeated iterative release of a “complete enough” deliverables for some use to be made of them by the customer.
- For those adopting agile methods, this can raise a key question / challenge:
  – How to produce a complete enough deliverable in each (time limited) sprint, for a complex system?
- Answer: Configured Patterns as draft deliverables—S*Patterns may be very quickly configured.

How the ASELCM Project will use the ASELCM Pattern

- The INCOSE ASELCM Project is a discovery project based upon data points emerging from workshops conducted with host enterprises during 2015-16:
  - Individual stories, experiences, challenges, and successes (with widely varying degrees of agility) can all be represented as “configurations” of the ASELCM Pattern.
  - The feedback loop in this process also contributes to refining the ASELCM Pattern—we are using the same approach.
  - All of which contributes to a technical reference model for the final project report.

Discussion

-
-
-
-
-
-
-

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. W. Schindel, "What Is the Smallest Model of a System?", *Proc. of the INCOSE 2011 International Symposium*, International Council on Systems Engineering (2011).
5. Bill Schindel, Troy Peterson, "Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques", in Proc. of INCOSE 2013 International Symposium, Tutorial, June, 2013.
6. W. Schindel, "Maps or Itineraries?: A Systems Engineering Insight from Ancient Navigators", to appear in Proc. of INCOSE International Symposium 2015, July, 2015.
7. ----------------, "System Life Cycle Trajectories: Tracking Innovation Paths Using System DNA", to appear in Proc. of INCOSE International Symposium 2015, July, 2015.
8. Schindel and Beihoff: "Systems of Innovation I: Models of Their Health and Pathologies", Proc. of INCOSE International Symposium, 2012.
9. 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.

Attachment 1

(Contains sample extracts from ASELCM Pattern)