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

Attachment 1

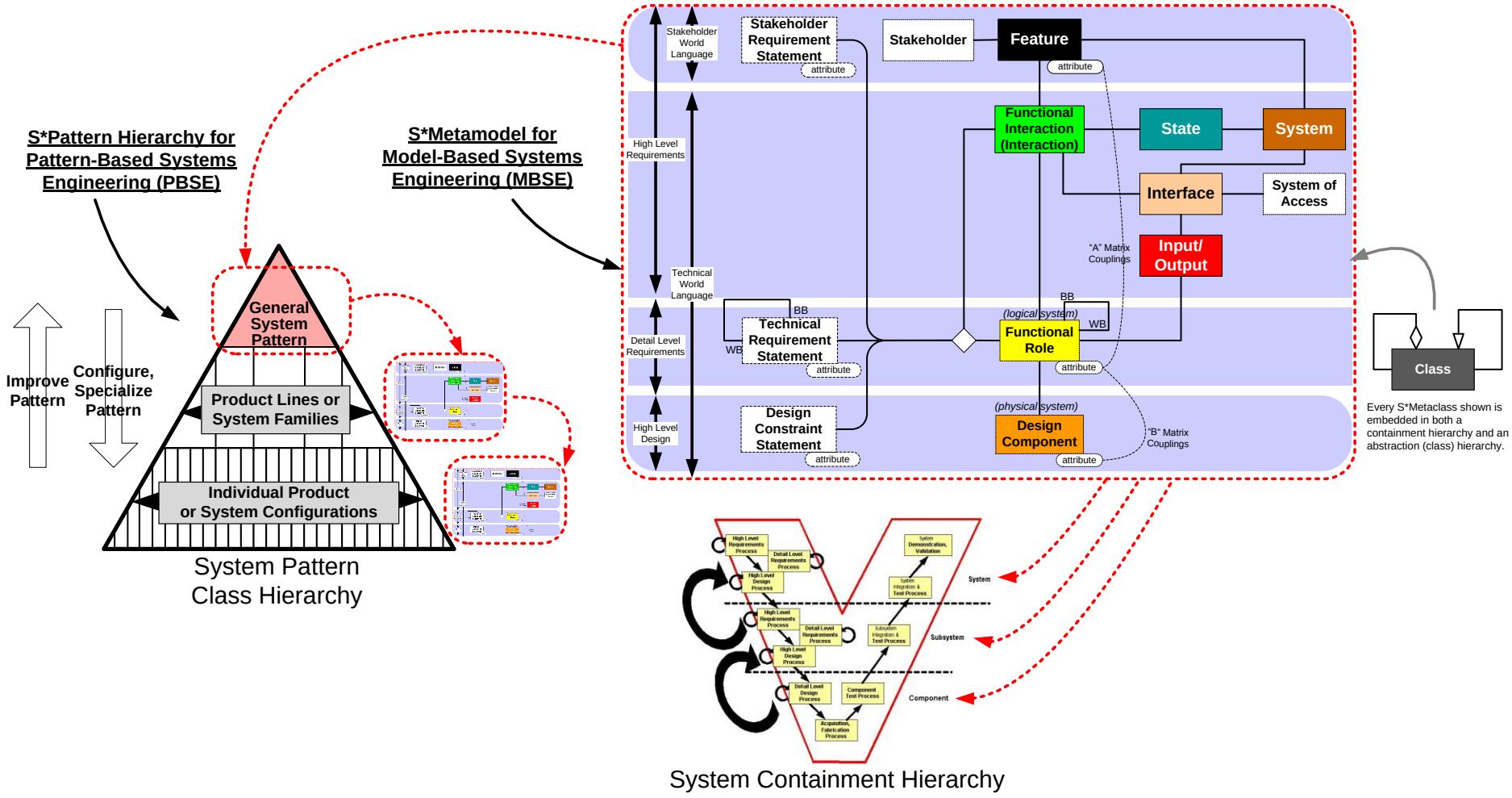
Sample Extracts from ASELCM Pattern

Bill Schindel, ICTT System Sciences
schindel@ictt.com

Attachment: Sample Extracts from ASELCM Pattern

- S*Metamodel Summary Extract
- ASELCM Domain and Logical Architecture Model Extracts
- ISO 15288 Feature and Logical Architecture Model Extracts
- Scrum State, Information, and Interaction (Activity Diagram) Model Extracts

S*Metamodel Summary Extract



Agile System Pattern Content

- Summary:
  - Domain Model (*)
  - Logical Architecture Model (*)
  - Physical Architecture Model
  - Input-Outputs, Interfaces, Systems of Access
  - Features Model (*)
  - Attribute Coupling Model
  - Interactions Model
  - State Model
  - Requirements Model

**** Items marked (*) are sampled with extracts here, from the overall ASELCM Pattern***

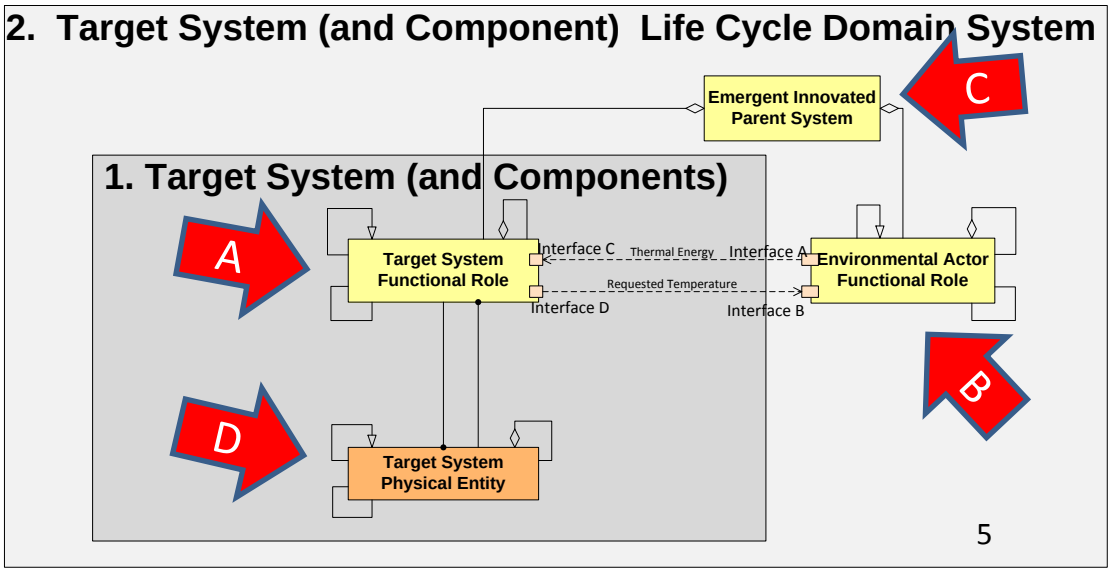
Logical Architecture and Physical Architecture of the Target System

A. The Innovated (Target) System is partitioned into a collection of **Target System Functional Roles.** 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 **Environmental Actor Functional Roles** played by environmental actors in the Target System (and Component) Life Cycle Domain System.

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

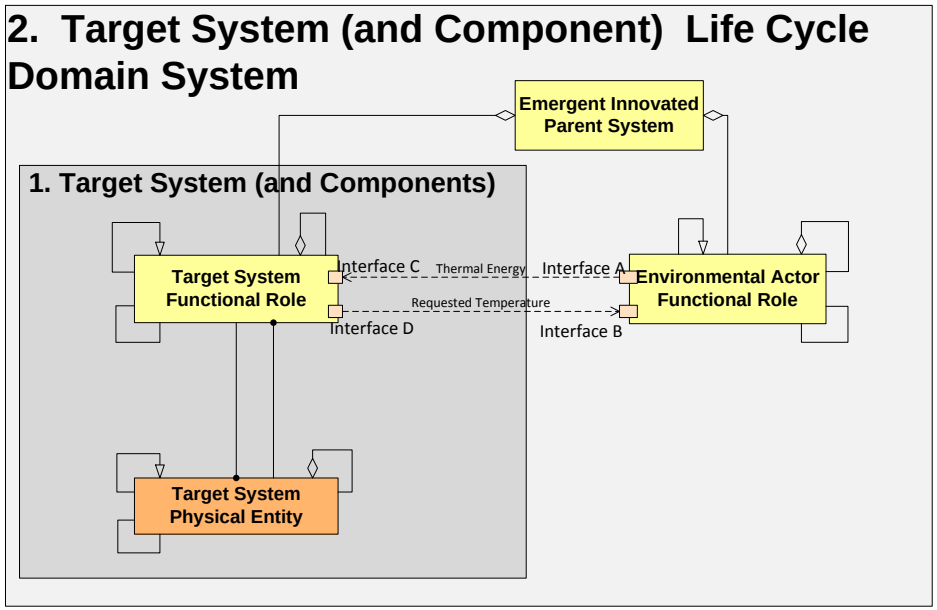
D. The Target System Functional Roles are allocated to **Target System Physical Entities** that perform those roles:
   - There can also be hierarchies of these.

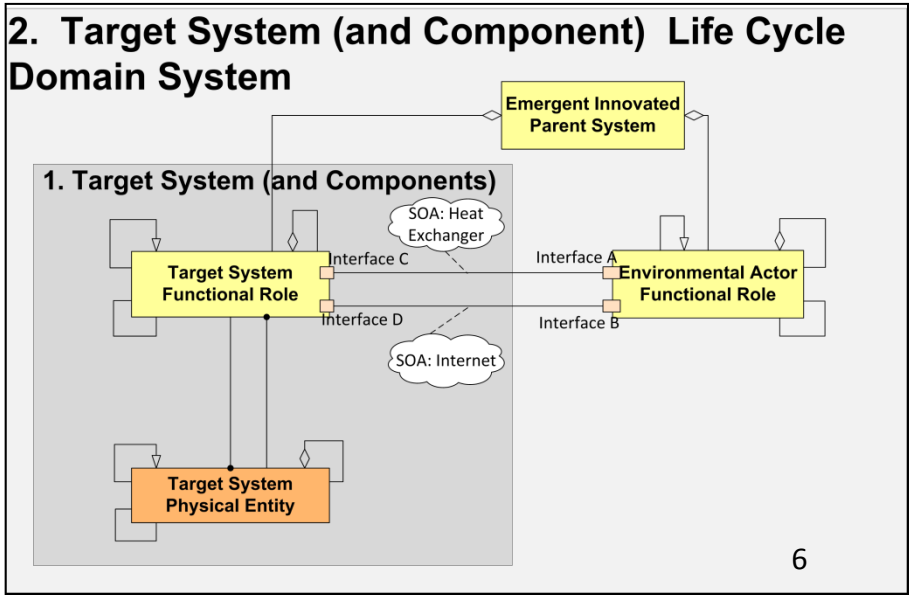


Logical Architecture and Physical Architecture of the Target System

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

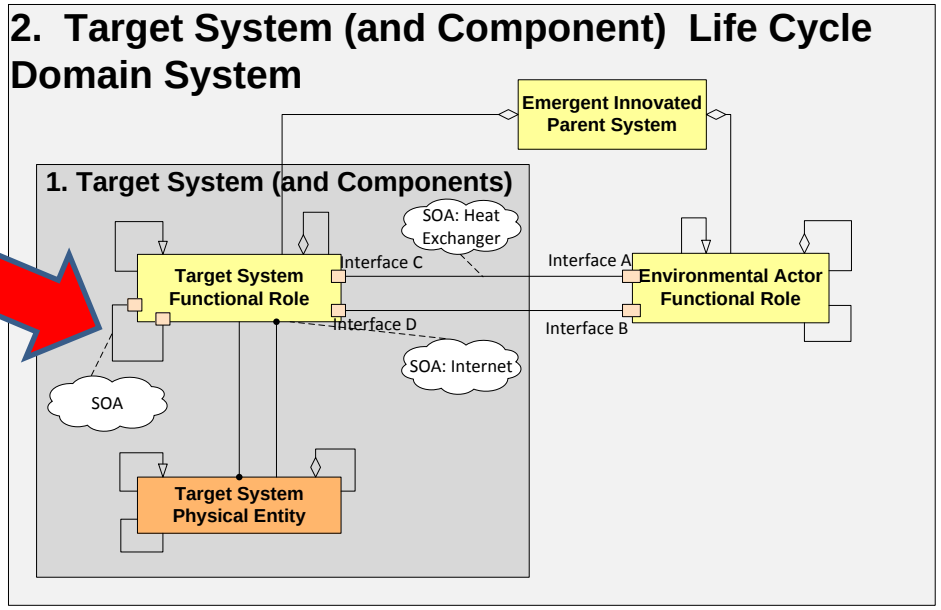
B. These interactions occur through **Interfaces**, which associate:
   - Systems, that "have" the Interfaces
   - Input-Outputs that pass through the Interface
   - Interactions which describe behavior at the Interface
   - **Systems of Access**, which are external intermediary "clouds" transporting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



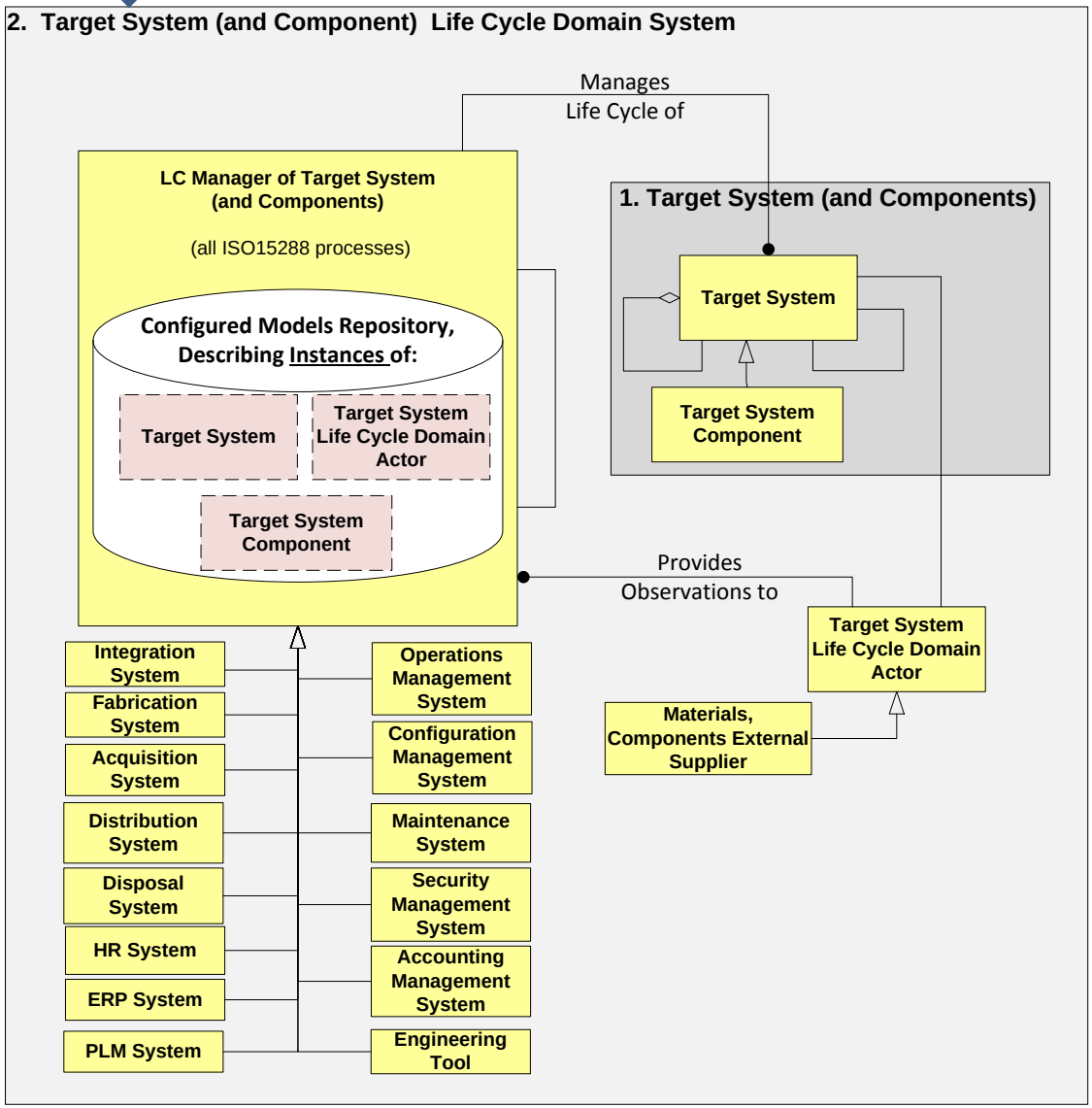


Logical Architecture and Physical Architecture of the Target System

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



Logical Architecture of
2. Target System (and Component) Life Cycle Domain System

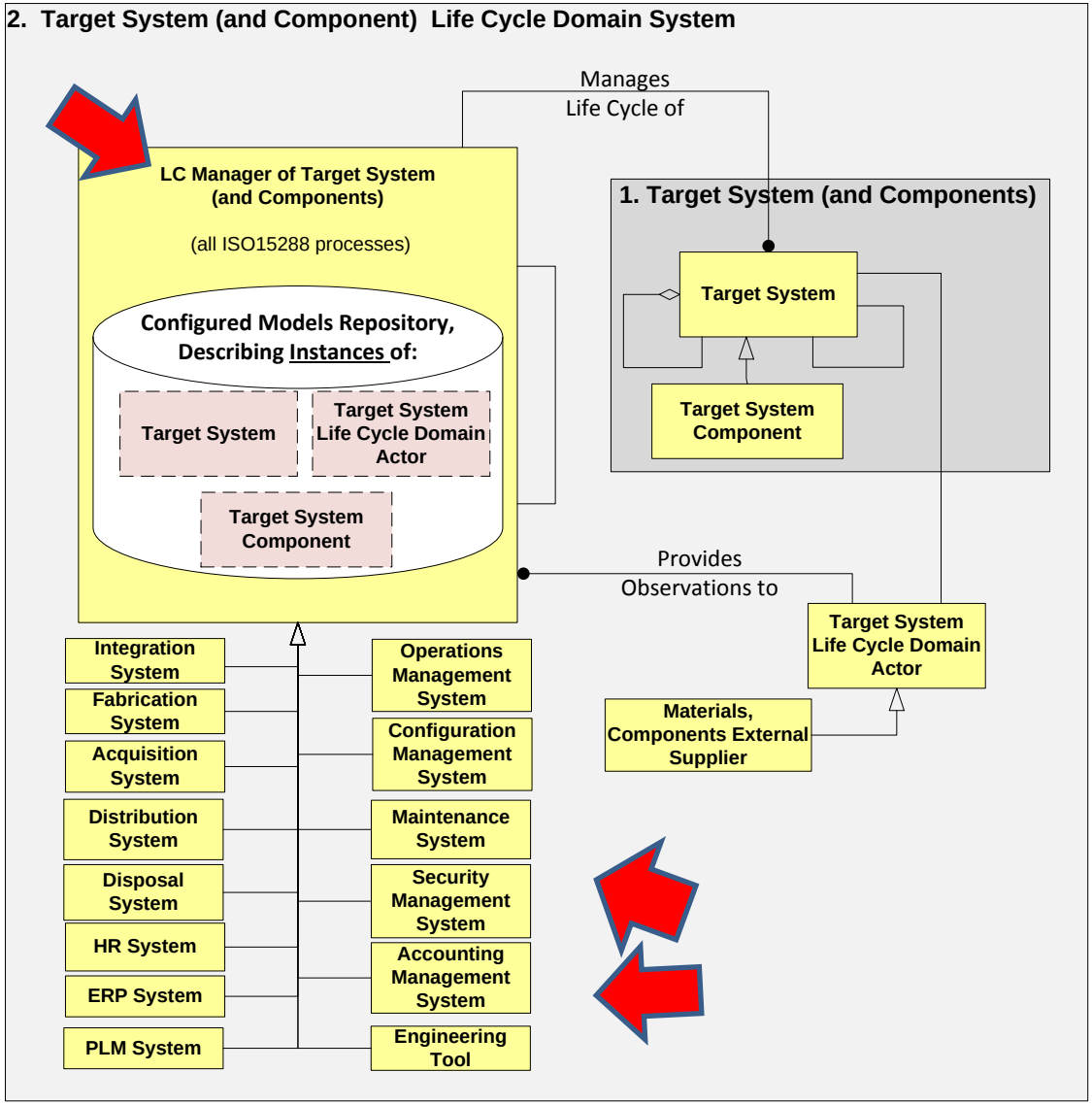


Definition: The logical system within which the Target System will exist during its life cycle, when "in service" or otherwise.

- This domain includes all actors with which the Target System will directly interact during its life cycle. This includes 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.

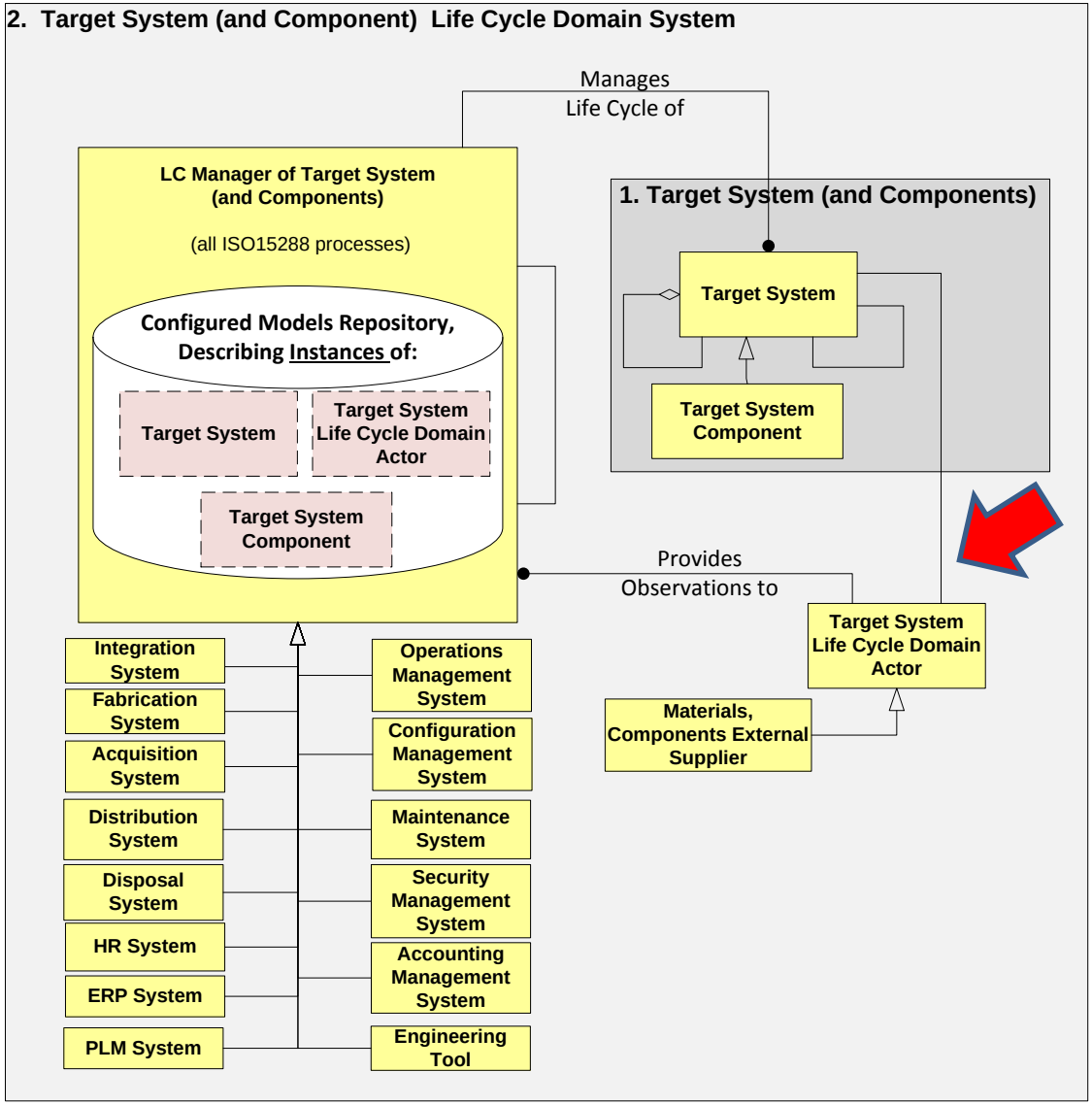
- The "disk drive" symbol represents potential information instance management, whether by electronic information technology, manual, biological, or other means.

LC Manager of Target System



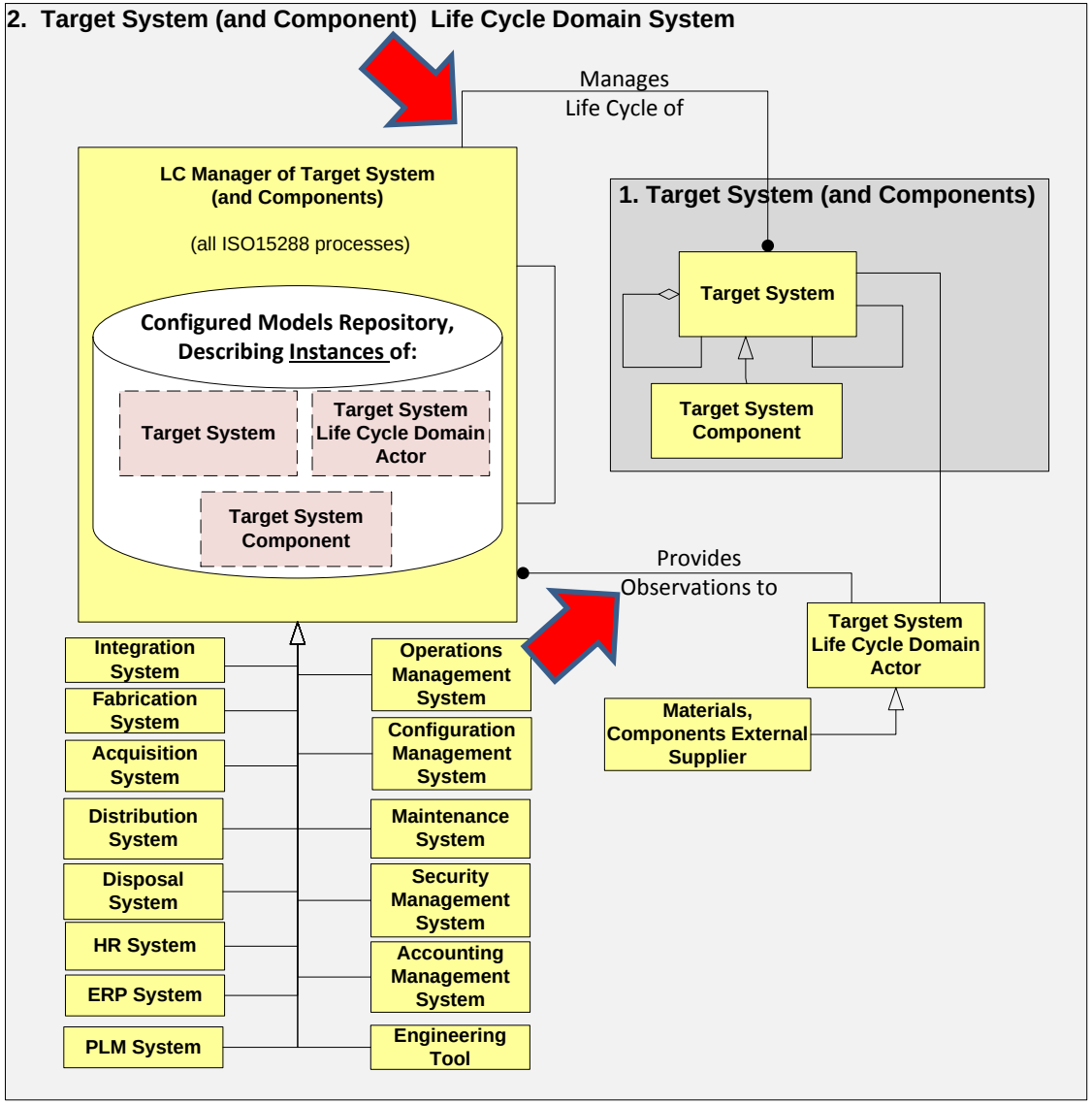
Definition: Manages Instances of Target Systems in Domain Environments, based on rules in Pattern; it includes not just model instance repository roles, but also the real behavior of managing the system in all its life cycle managed situations—including all ISO15288 processes; includes production [including decisions to produce quantities based on need], acquisition of needed quantities of externally supplied entities and services, distribution [including marketing], operation, maintenance, configuring, securing, accounting, and disposal; these also include configuring and storing models of the managed instances; it covers not just target systems, but also the components eligible to become parts of them.

Target System Life Cycle Domain Actor



Any system with which the Target System (or Target System Component) directly interacts during its life cycle (except for the LC Managers). Includes all environmental actors encountered by the "in service" Target System, as well as non-management actors in the rest of the Target System life cycle.

Collection of Observations for Future Learning

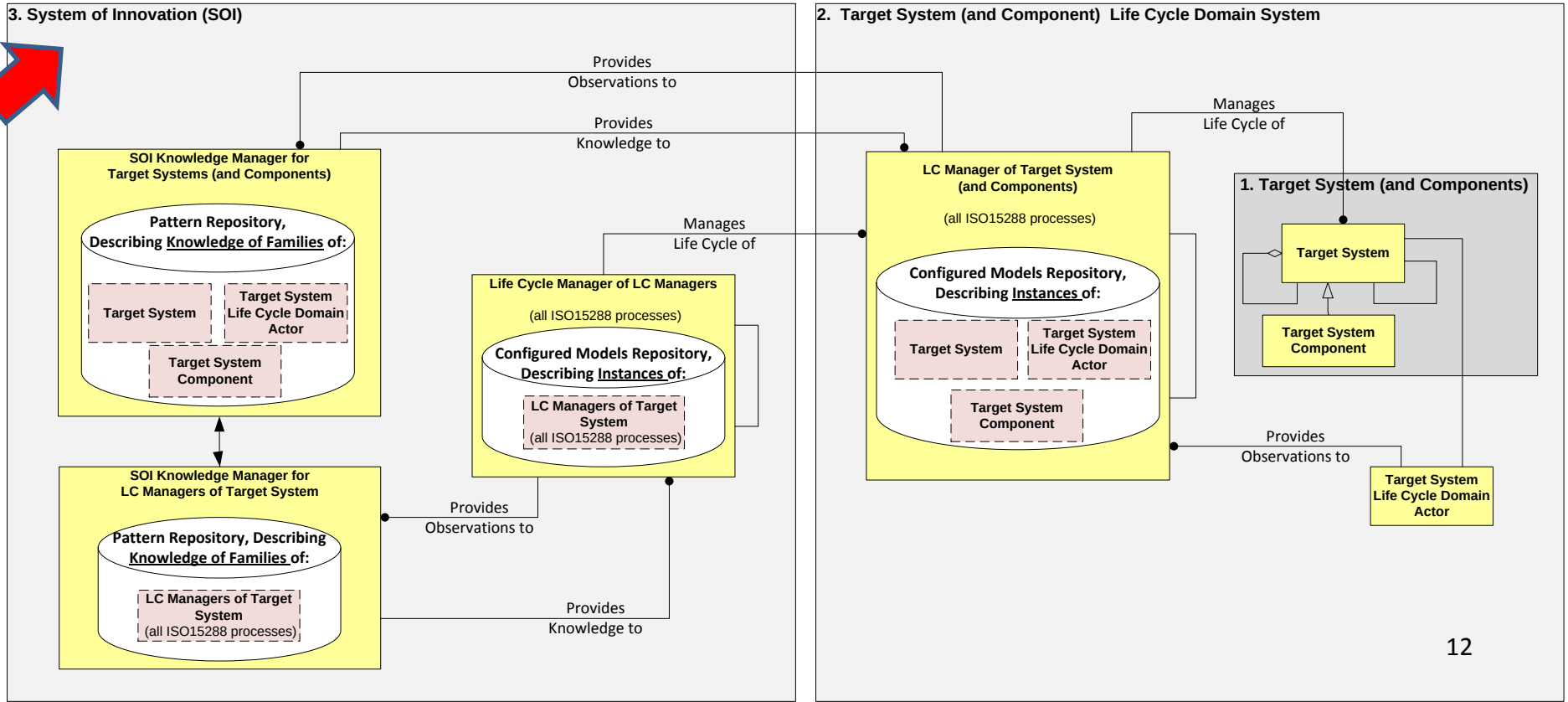


The LC Manager of Target System (and Components) is also responsible for collecting “observations” (telemetry, etc.) about the Target System and its Environment.

These observations will later be provided to the System of Innovation, for its analysis and distillation, as a part of Knowledge Management.

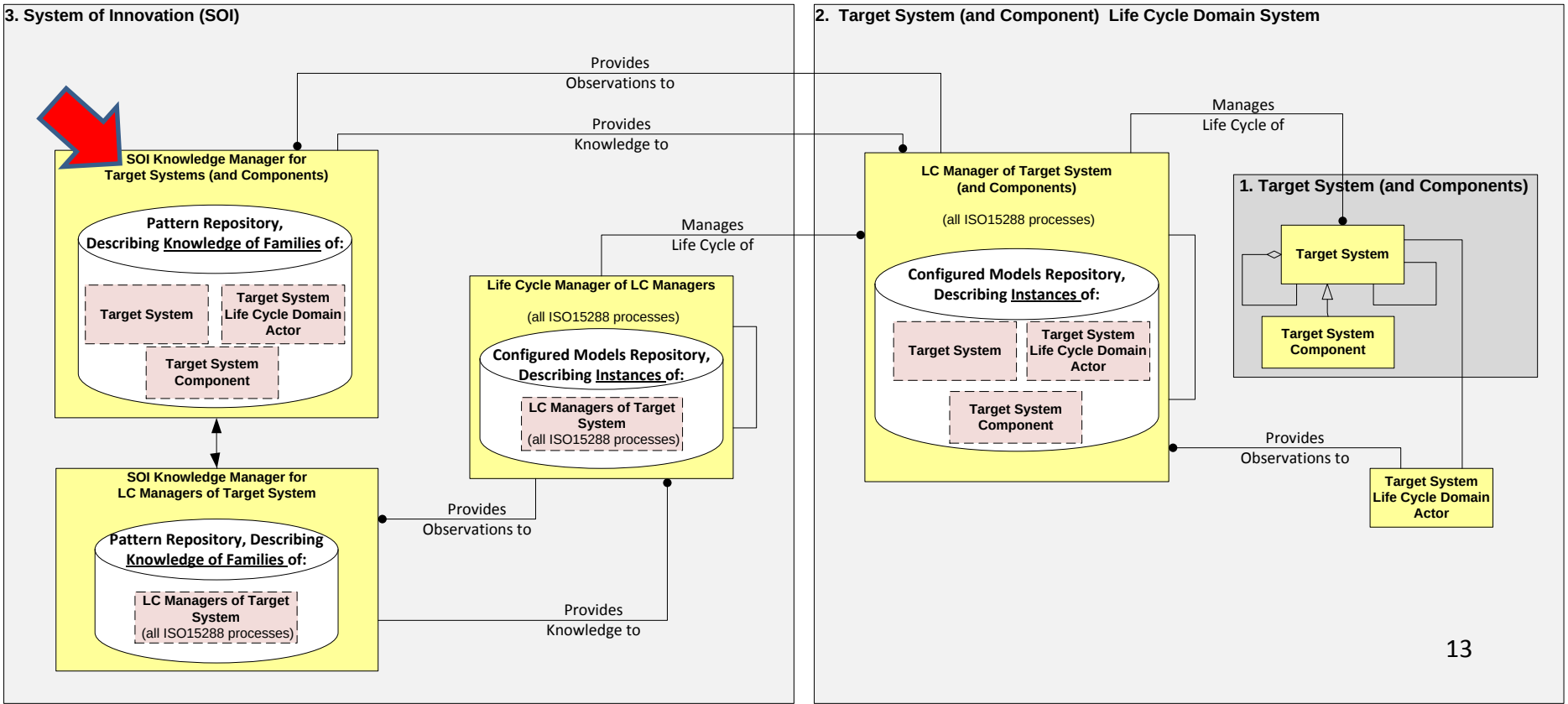
System 3: The System of Innovation

- Definition: 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.



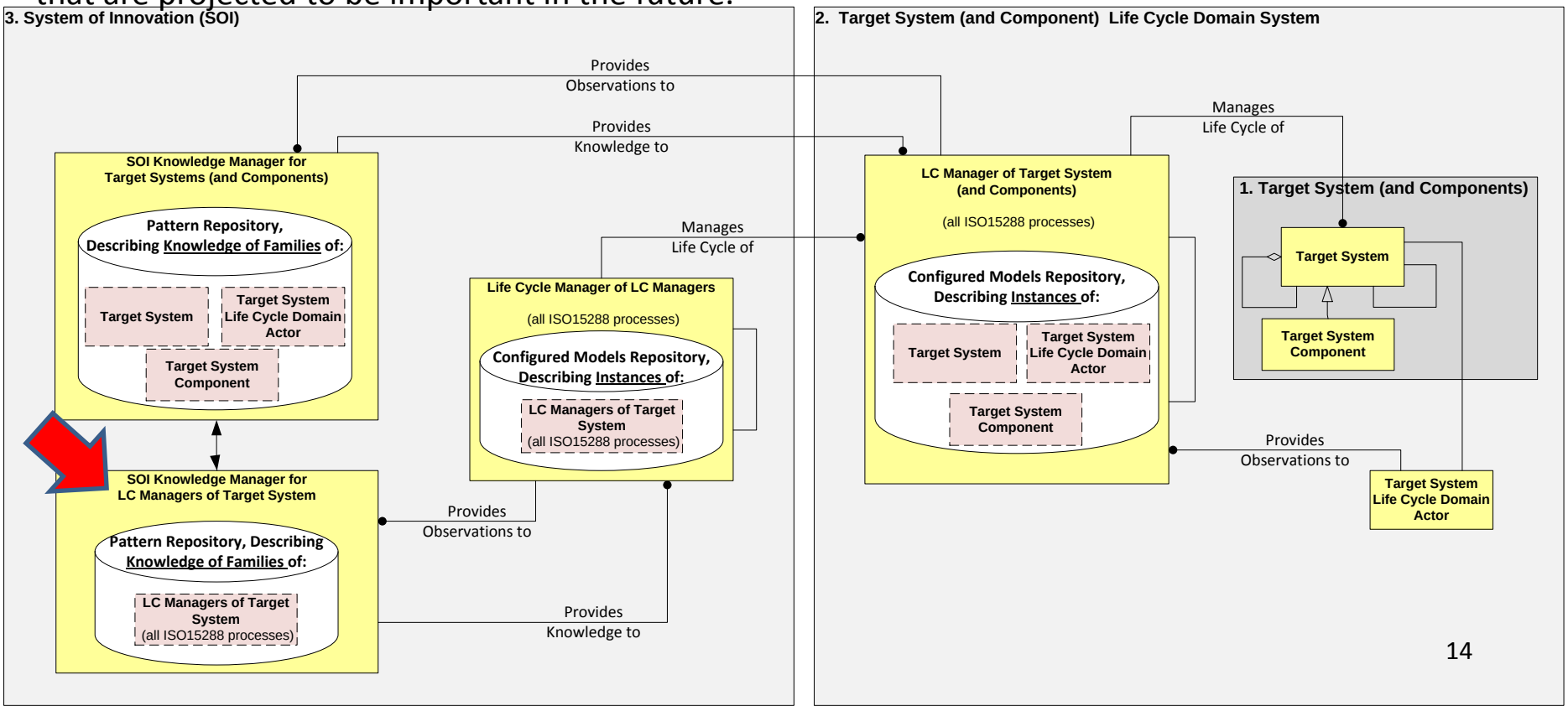
SOI Knowledge Manager for Target System (and Components):

- Definition: The logical system responsible for managing general knowledge (patterns, rules, general models, etc.) of Life Cycles of classes of Target Systems (and their Components) in Domain Environments.
- This includes creating, updating, and maintaining configurable knowledge patterns, including indications of confidence in aspects of the model, for use by the LC Manager of Target System.
- It also includes comparing observations of instances of real target and domain systems to the model, whether by planned experiment or otherwise, and distilling learning updates from those comparisons.
- It also includes identifying what parts of the pattern are priority to fill in or verify, based on stakeholder interests, and planning experiments or what to observe; this includes identifying proactively aspects that are projected to be important in the future.



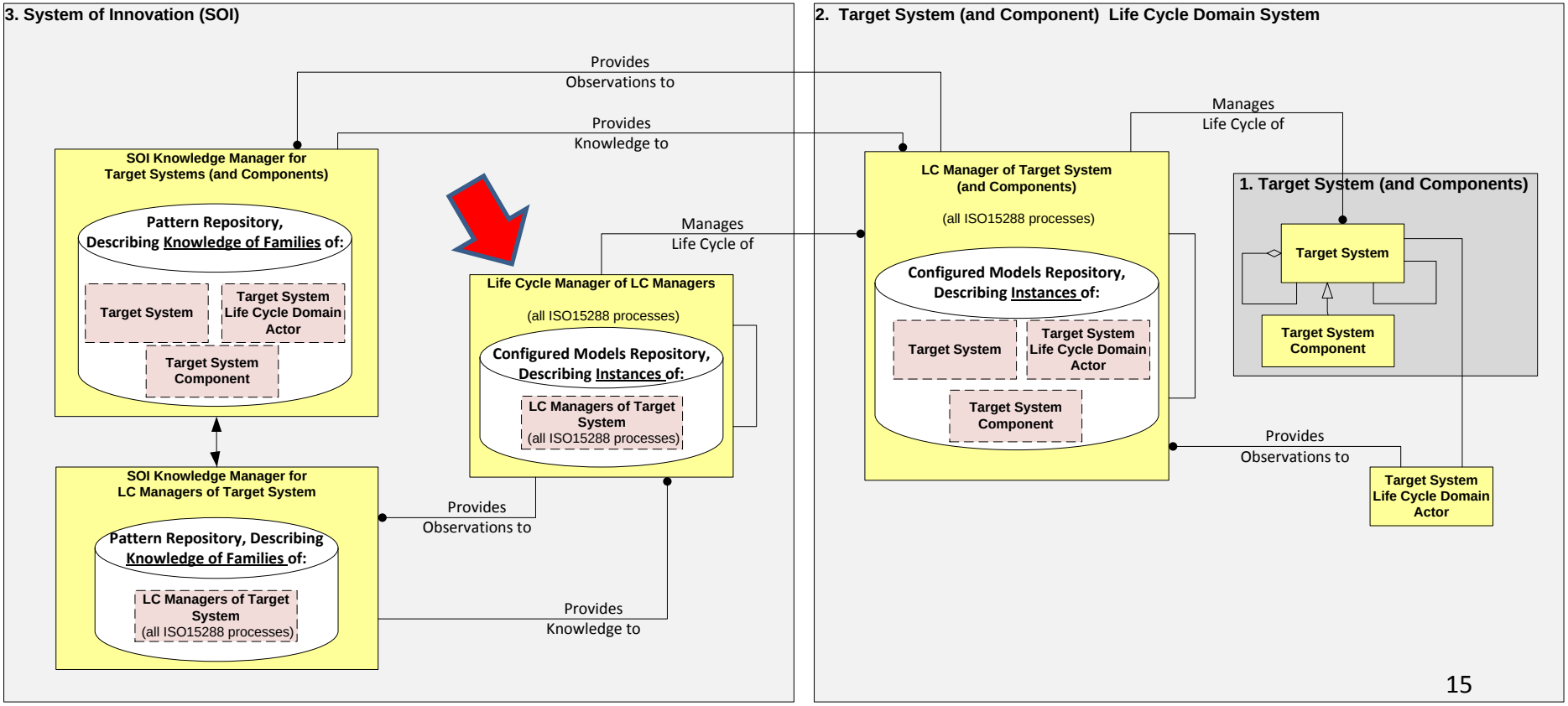
SOI Knowledge Manager for LC Managers of Target Systems:

- Definition: The logical system responsible for managing general knowledge (patterns, rules, general models, etc.) of Life Cycles of classes of LC Managers of Target Systems. (For example, general rules concerning life cycles of manufacturing systems; rules for changing Security Management Systems, etc.)
- This includes creating, updating, and maintaining configurable knowledge patterns, including indications of confidence in aspects of the model, for use by the LC Manager of LC Managers.
- It also includes comparing observations of instances of real LC Managers to the model, whether by planned experiment or otherwise, and distilling learning updates from those comparisons.
- It also includes identifying what parts of the pattern are priority to fill in or verify, based on stakeholder interests, and planning experiments or what to observe; this includes identifying proactively aspects that are projected to be important in the future.



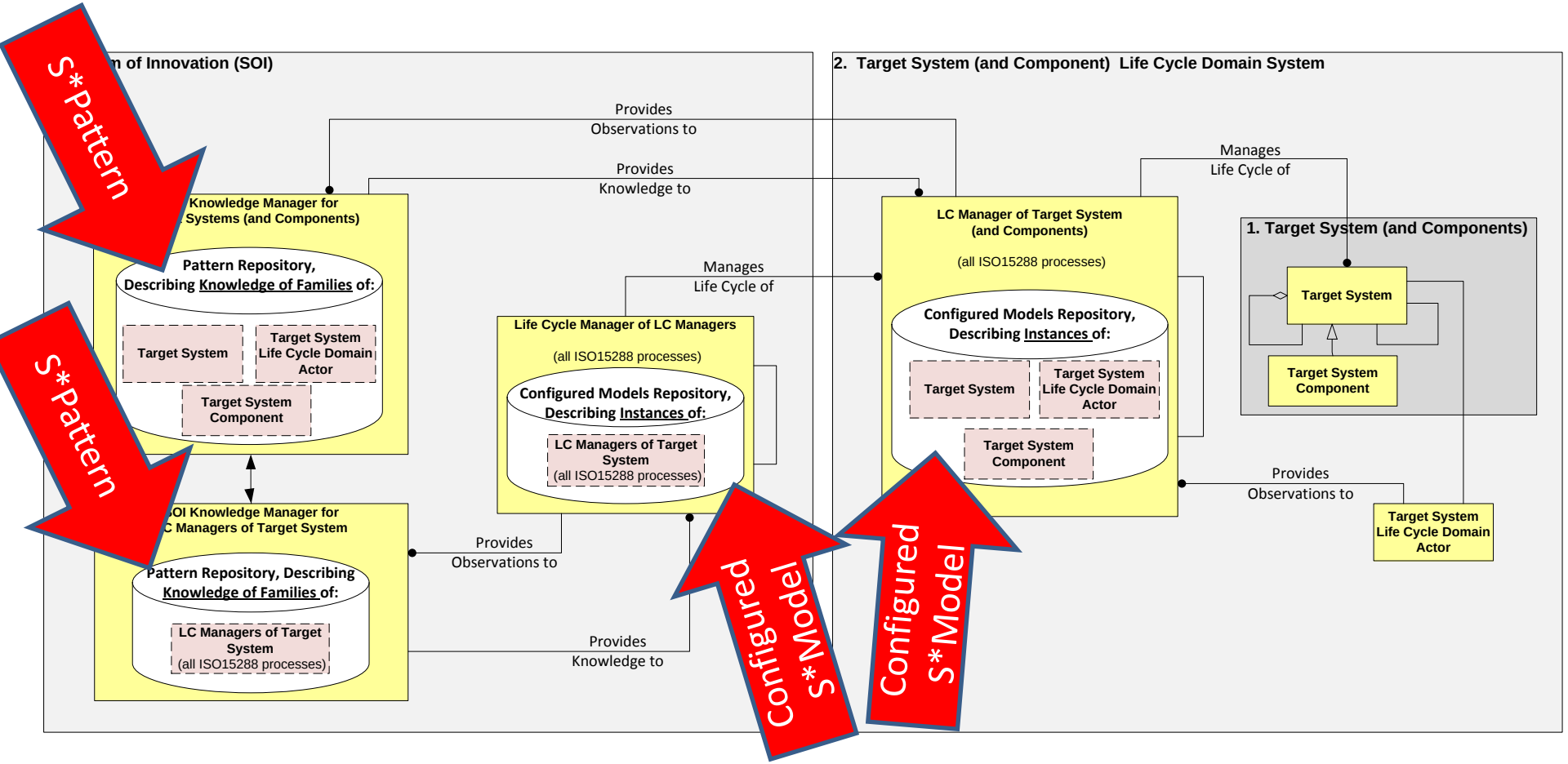
Life Cycle Manager of LC Managers:

- Definition: Manages all the Target System Life Cycle Manager instances, based on rules from Patterns. It includes storing models of the instances, as well as performing all (ISO 15288) life cycle management of those systems. (For example, engineering and installing a new production system for Target Systems.)
- Also responsible for collecting "observations" (telemetry, etc.) about the LC Managers of Target Systems. These observations are provided to the SOI Knowledge Manager for LC Managers, for its analysis and distillation, as a part of Knowledge Management.



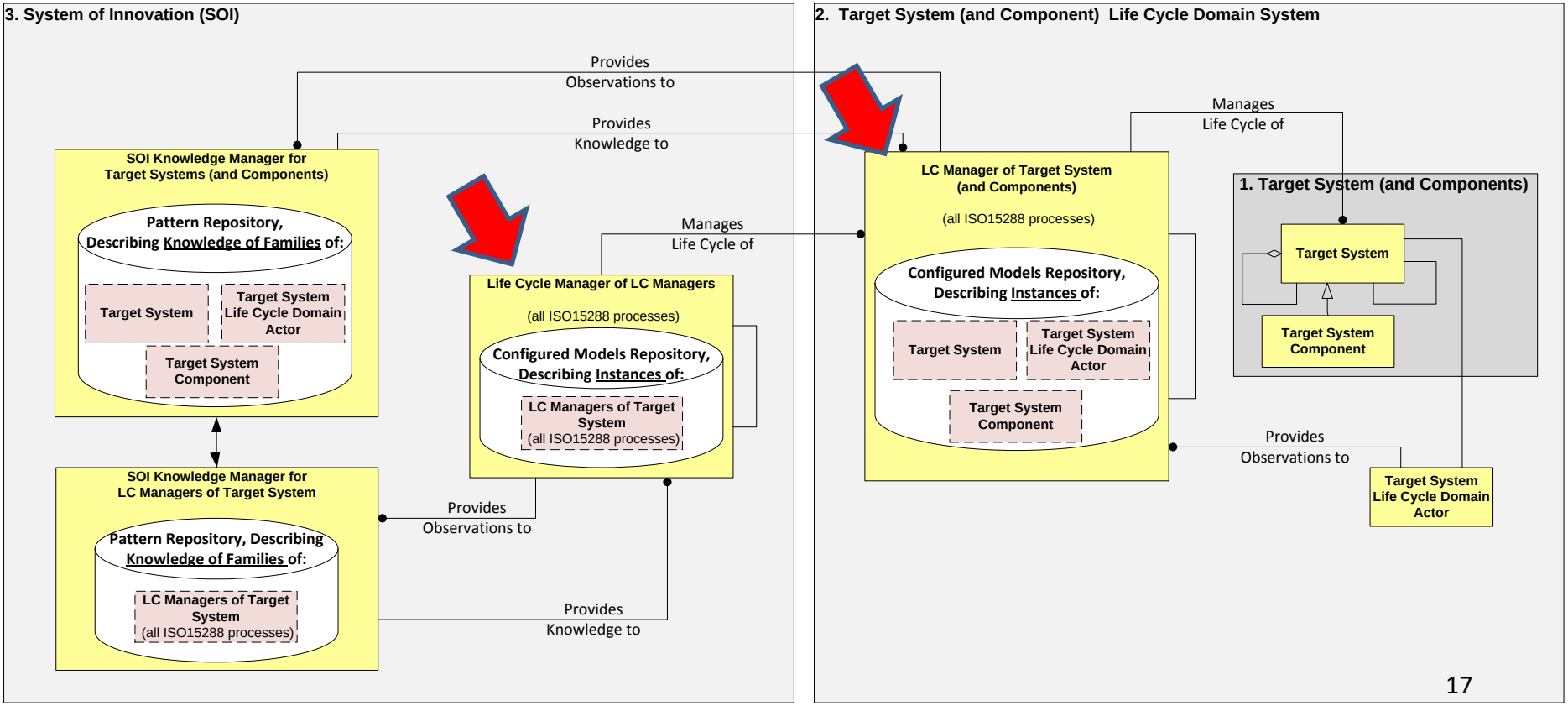
Knowledge Management, Models, and Patterns

- S*Models contain the minimum information needed, for purposes of describing life cycle managed systems.
- S*Models are configured from S*Patterns--re-usable, configurable S*Models, describing families of managed systems over various life cycle situations.
- For these purposes, Knowledge Management becomes Pattern Management.



Logical Architecture of the Life Cycle Managers:

- There are two overall Life Cycle Manager classes, responsible to manage very different systems.
- However, they have both been modeled using the ISO15288 reference model.
- In that sense, they can be said to share a common “reference” architecture . . .

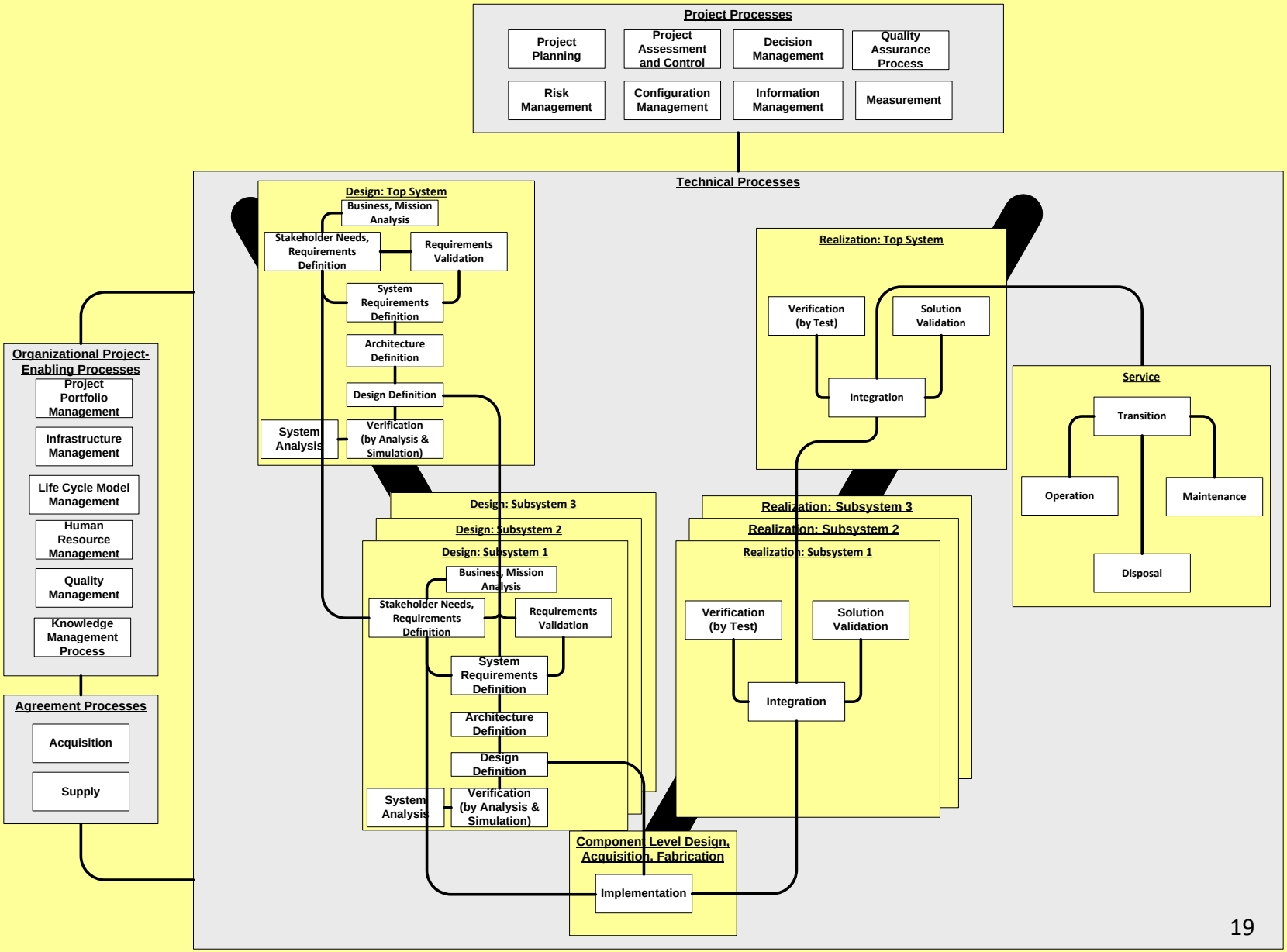


ISO 15288 Model Extracts

- Logical Architecture
- Feature Model

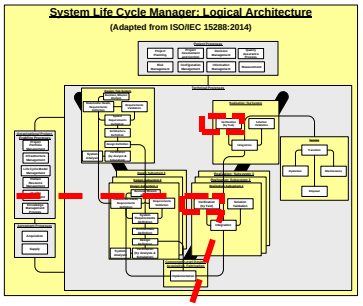
System Life Cycle Manager: Logical Architecture

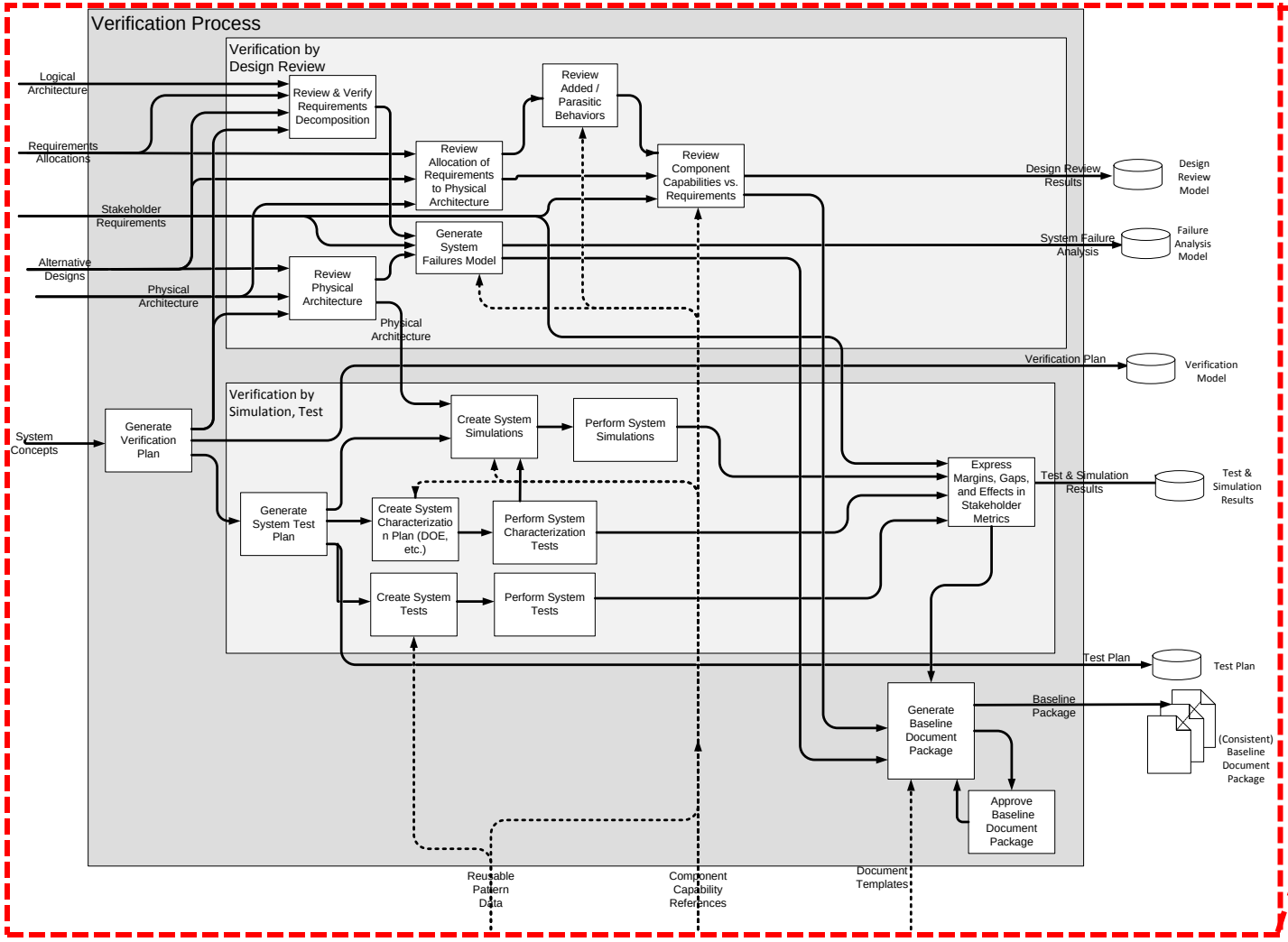
(Adapted from ISO/IEC 15288:2014)



A model is provided for each of the ISO15288 Processes

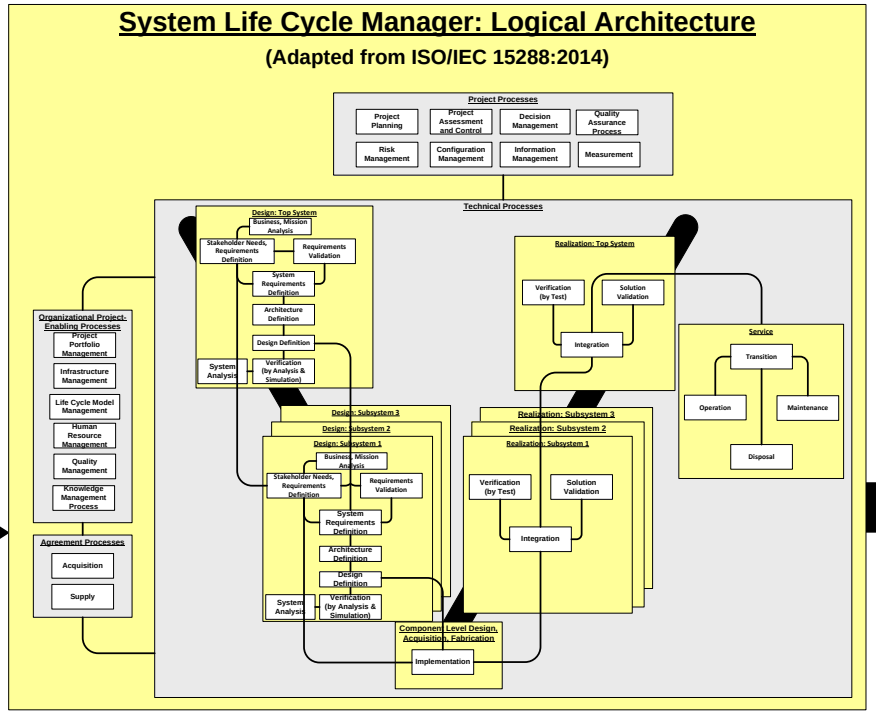
- Example: Verification Process
- Each provides options for MBSE and PBSE methods

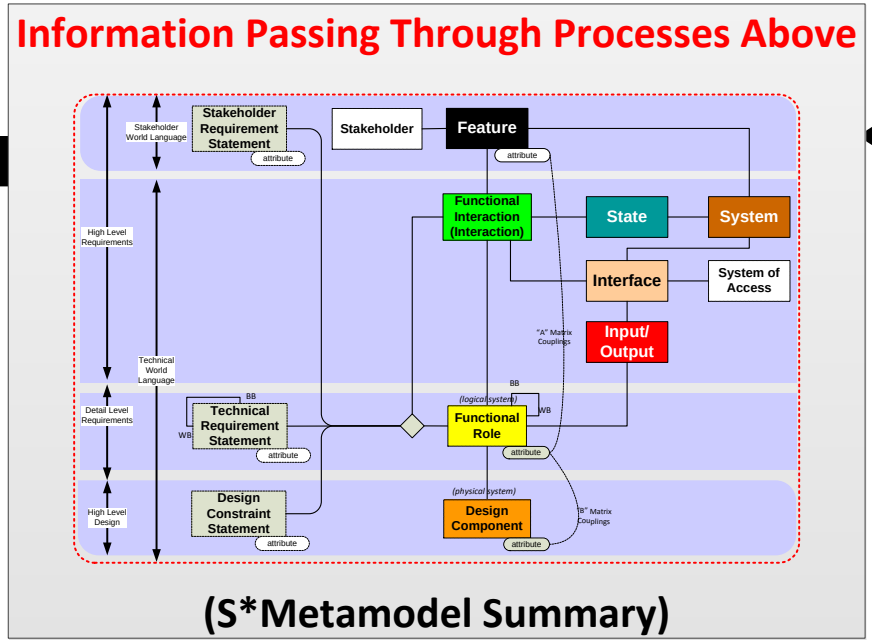




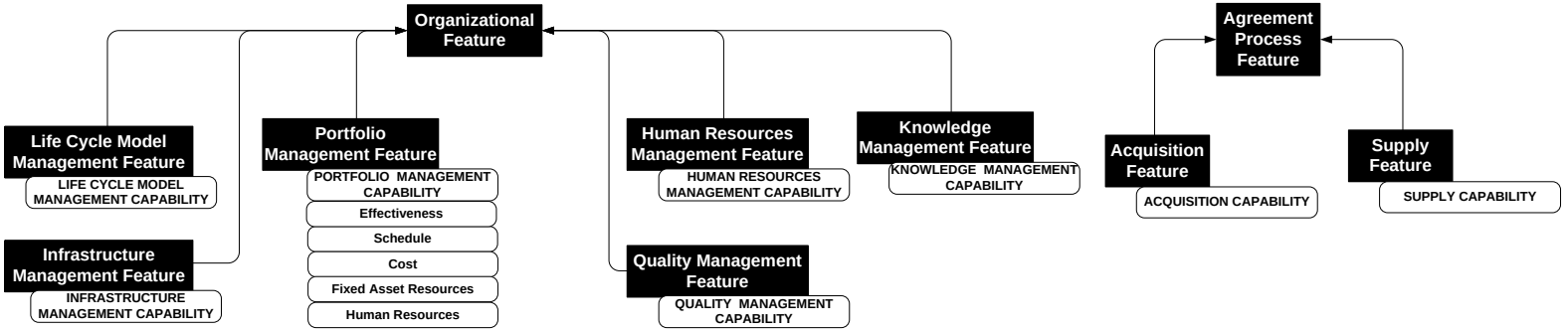
Process vs. Information:

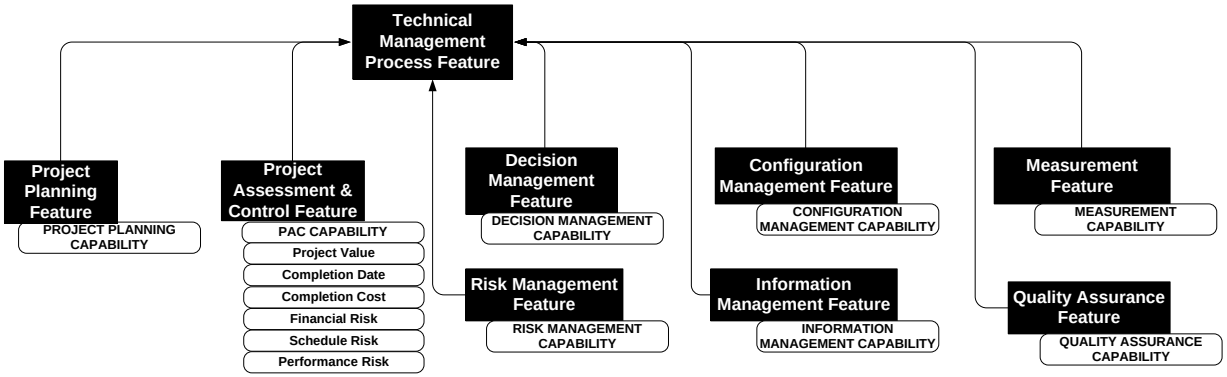
- The S*Metamodel describes the MBSE information that passes through life cycle processes, as S*Models.
- Using PBSE accelerates this process, by basing S*Model information on knowledge-managed S*Patterns.
- None of this requires any specific sequence or order of processes, which may be concurrent or otherwise, depending on strategy.
- What is the “agile trajectory” through S*Space?
- Agile Scrum strategy is to “sprint” short distances in S*Space, with fixed time and resource budgets.
- How are the “trajectory deltas” planned for each Agile Scrum?

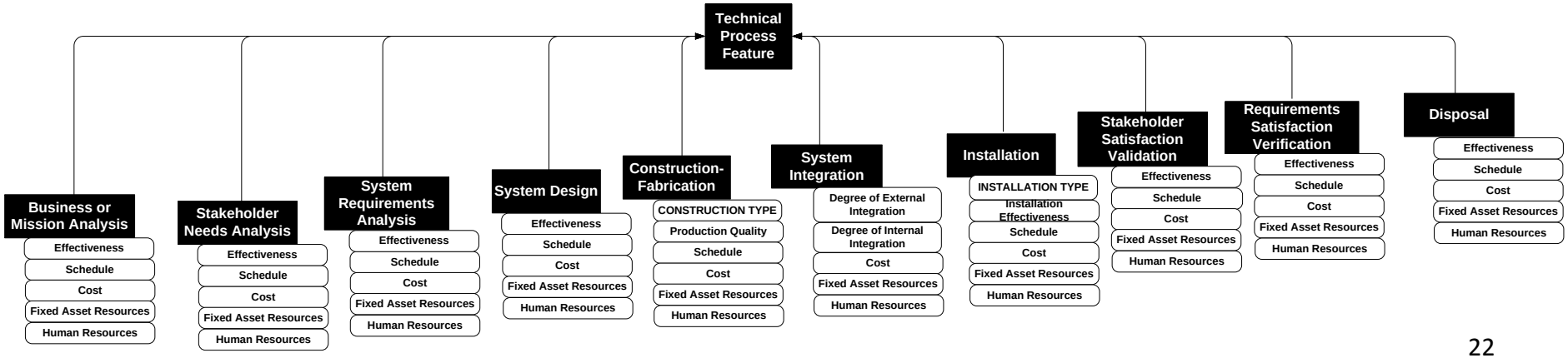




Overview of S*15288 Pattern Features



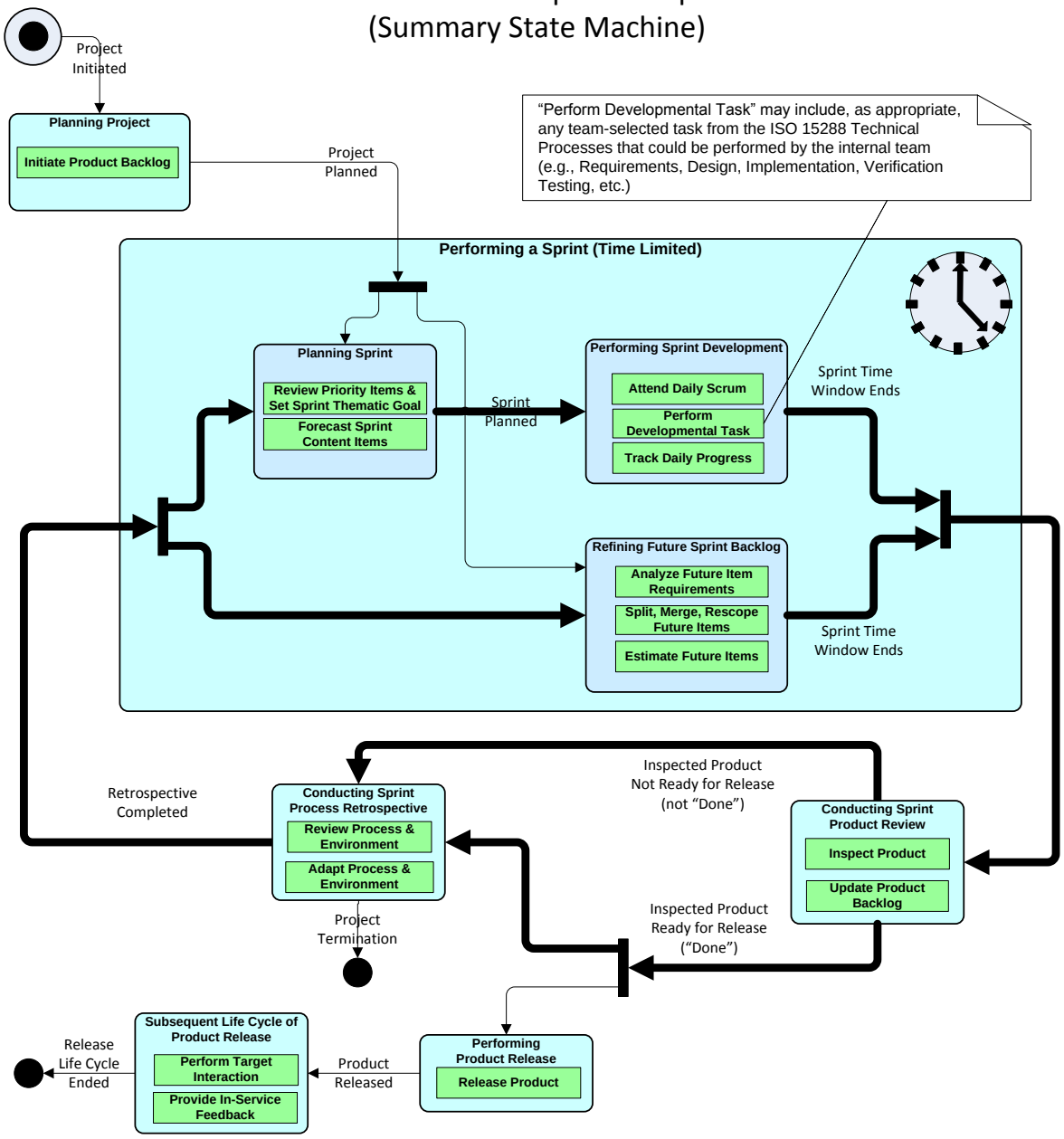




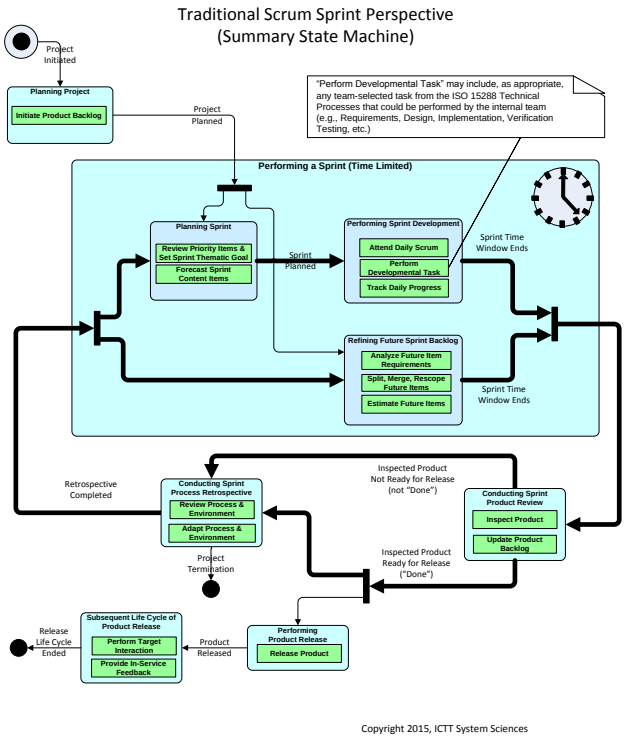
Scrum Model Extracts

- Summary State Model
- Managed Information Model
- Activity Diagram

Traditional Scrum Sprint Perspective (Summary State Machine)



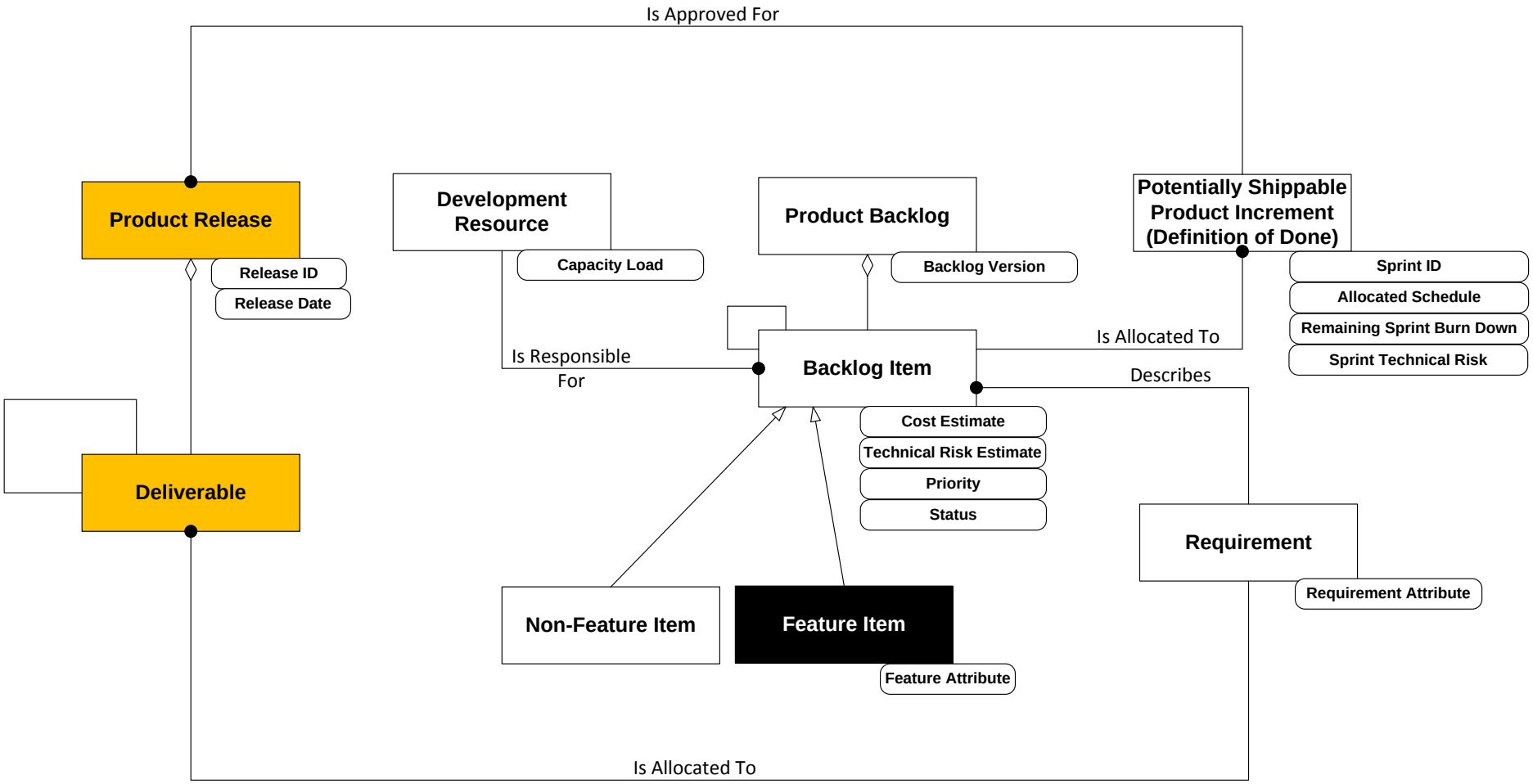
Copyright 2015, ICTT System Sciences



Copyright 2015, ICTT System Sciences

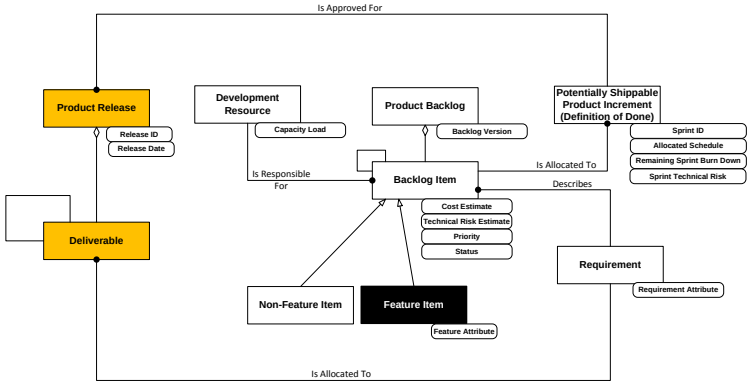
- This diagram is one way to represent traditional agile development, emphasizing in this view the “states” that the process passes through, and the “interactions” that occur during those states.
- The leading feature of agile development is that it is performed in a series of (short) defined time increments, instead of defined content increments.
- These time increments are called “sprints”, in recognition of their short (not more than 30 days) length.
- Content of these increments is nevertheless still prioritized and planned (important!), but in actual execution it is the elapse of time which signals end of a sprint, and not the completion of the planned work.
- Planned deliverable content of each sprint is chosen so that, if achieved, it would be complete enough to incrementally deliver to the customer—including sufficiently tested, documented, etc. , even though not yet complete as a whole project yet.
- After each sprint, inspection of the incremental delivered product in (at least simulated) “operational use”, along with internal observation of the development process, are used to generate feedback for subsequent sprints.
- The agile process is thus inherently small step exploratory, instead of a top-down large commitment “cliff”—a strategy recognizing inherently unfamiliar, unknown, dynamically changing, or otherwise risky situations.

Traditional Scrum Sprint Perspective (Simplified Model of Managed Information)



Copyright 2015, ICTT System Sciences

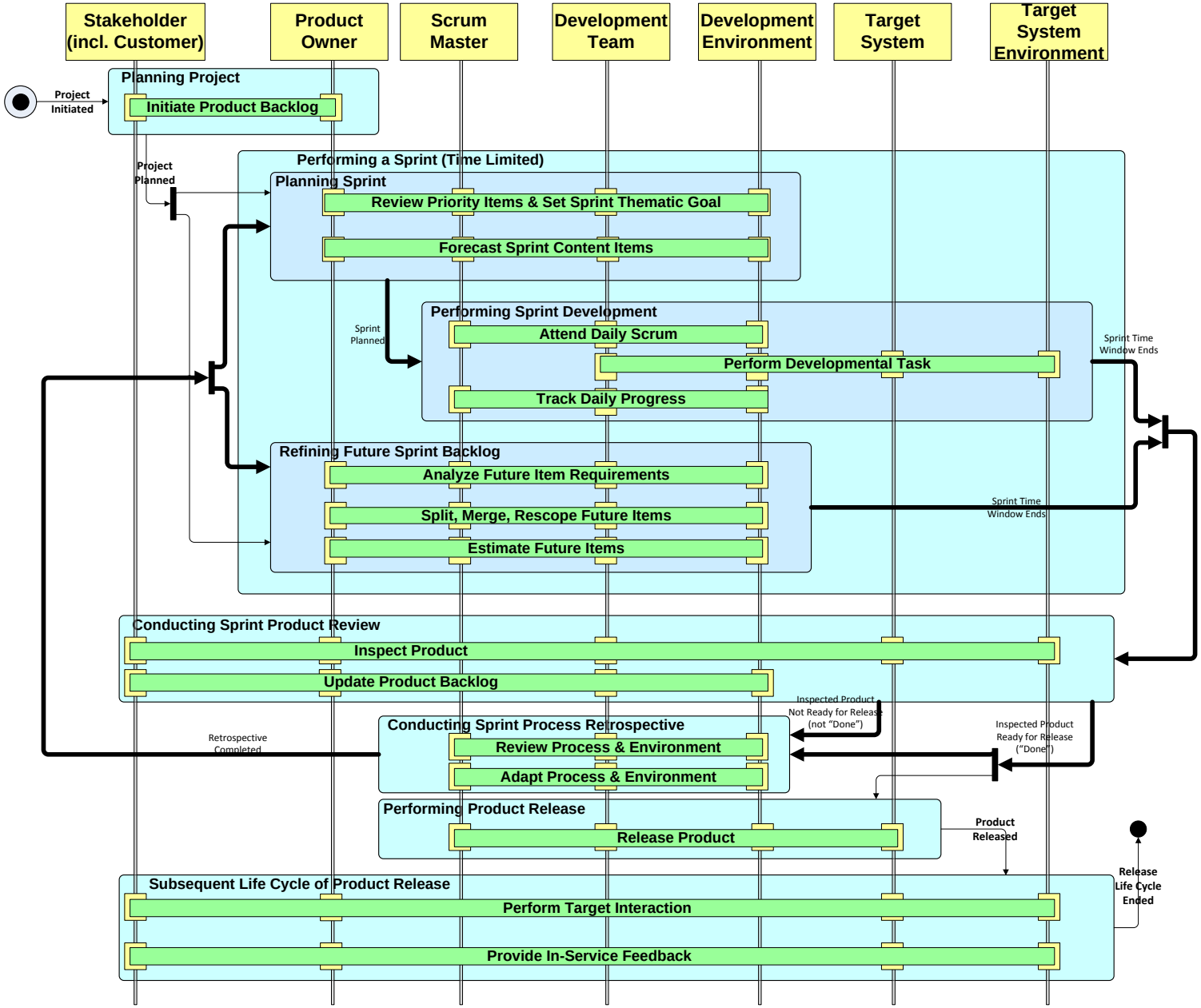
Traditional Scrum Sprint Perspective
(Simplified Model of Managed Information)



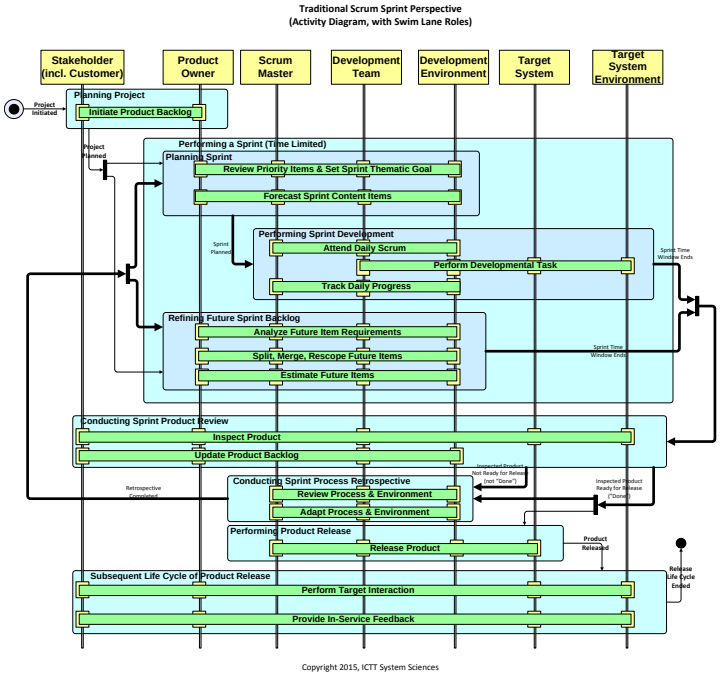
Copyright 2015, ICTT System Sciences

- Traditional agile methods focus on certain key information, summarized by this information model:
- The Product Backlog is a list of Backlog Items that are candidates for inclusion in deliveries.
- Backlog items may be Features for stakeholders, or other (internally prioritized) non-Feature items. (In S*Patterns, the stakeholder model is wide enough that there really are no needs that are not for some stakeholder, so the Feature items should cover everything.)
- Backlog Items are prioritized by the Product Owner, interacting with the Customer (and other stakeholders).
- Backlog Items are analyzed by the Development Team to produce more descriptive Requirements, along with estimates of related cost or effort.
- Since Deliverables can (as appropriate) include Design, Test Results, etc., there can be more ISO15288 content in this simple model than first appears.
- Based on priorities and estimates Backlog Items are chosen to include in a Sprint.
- During the Sprint, team members record daily progress, including updates to how much effort will apparently be needed to complete the sprint, tracked in the Sprint Burndown

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

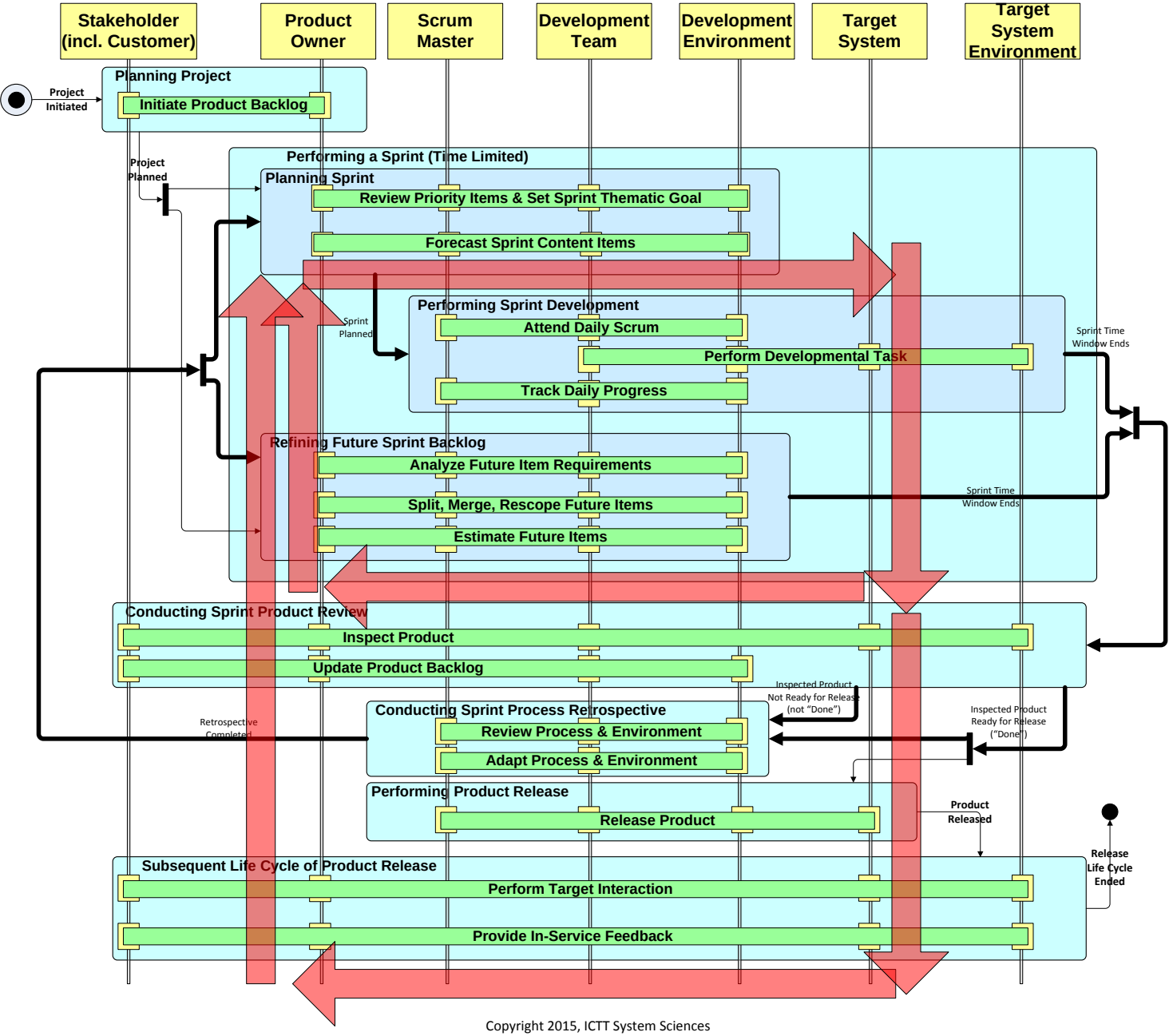


Copyright 2015, ICTT System Sciences



- This diagram represents the same agile process (notice the same process States and Interactions), but emphasizes the key roles associated with agile methods, and who is responsible to perform them—both development roles and actors they encounter:
- The Customer (more generally, all the stakeholder classes)
- The Product Owner, who represents all the stakeholders and is responsible for the value delivered to them, beginning with the prioritized Product Backlog (listing of what is to be delivered, in stakeholder Feature terms)
- The Development Team, intentional very small (~7 or fewer), and responsible for all the technical work necessary to deliver the product, including (as needed) requirements analysis, design, acquisition, construction, test/verification, etc.
- The Scrum Master, who serves the Development Team to make their work possible and productive.
- The Development Environment, including laboratory or other technical facilities, test stands, and the general work environment.
- The Target System to be developed or modified by the delivery process.
- The Target System Environment, which it will encounter over its subsequent life cycle.

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



Copyright 2015, ICTT System Sciences