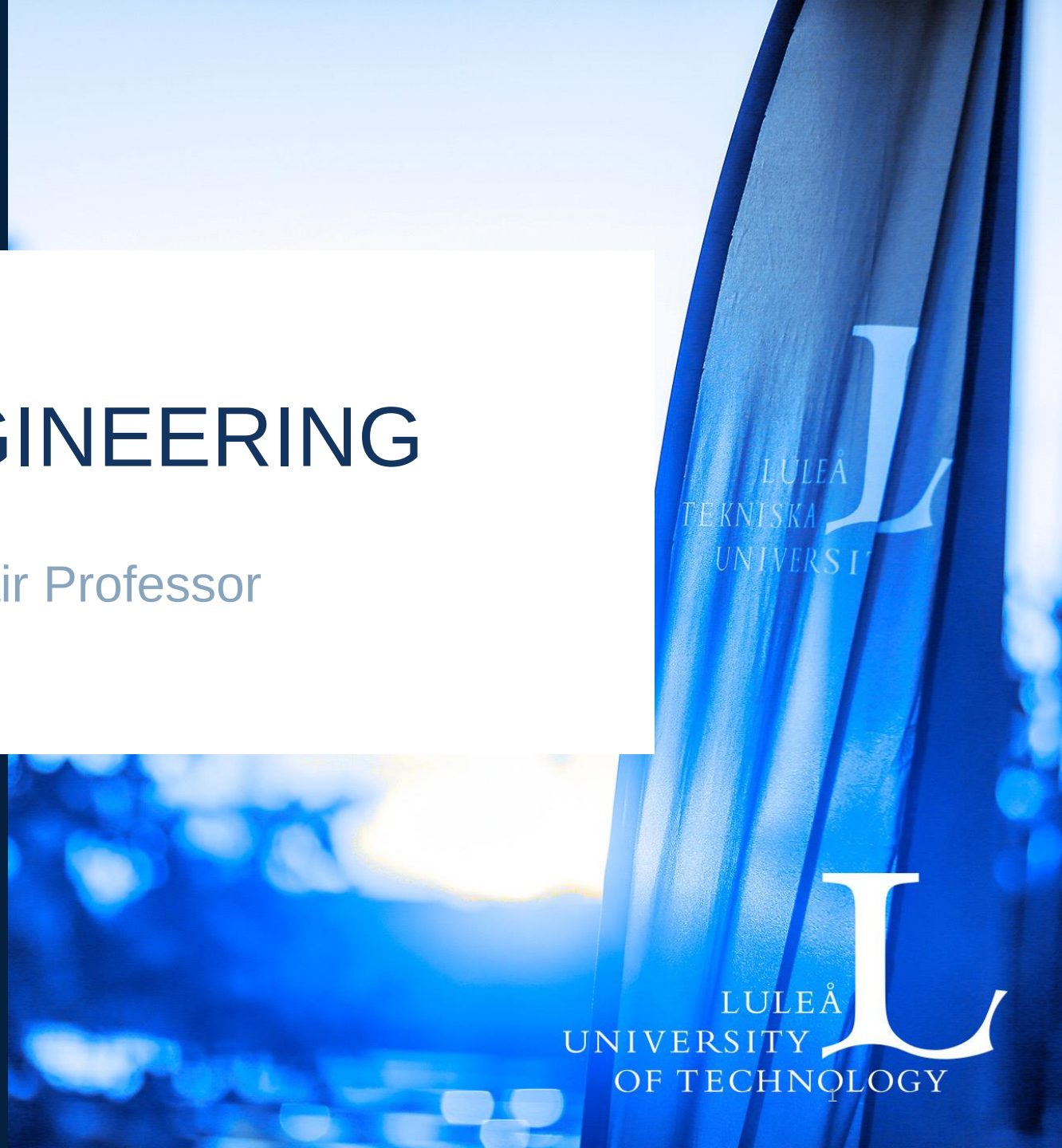
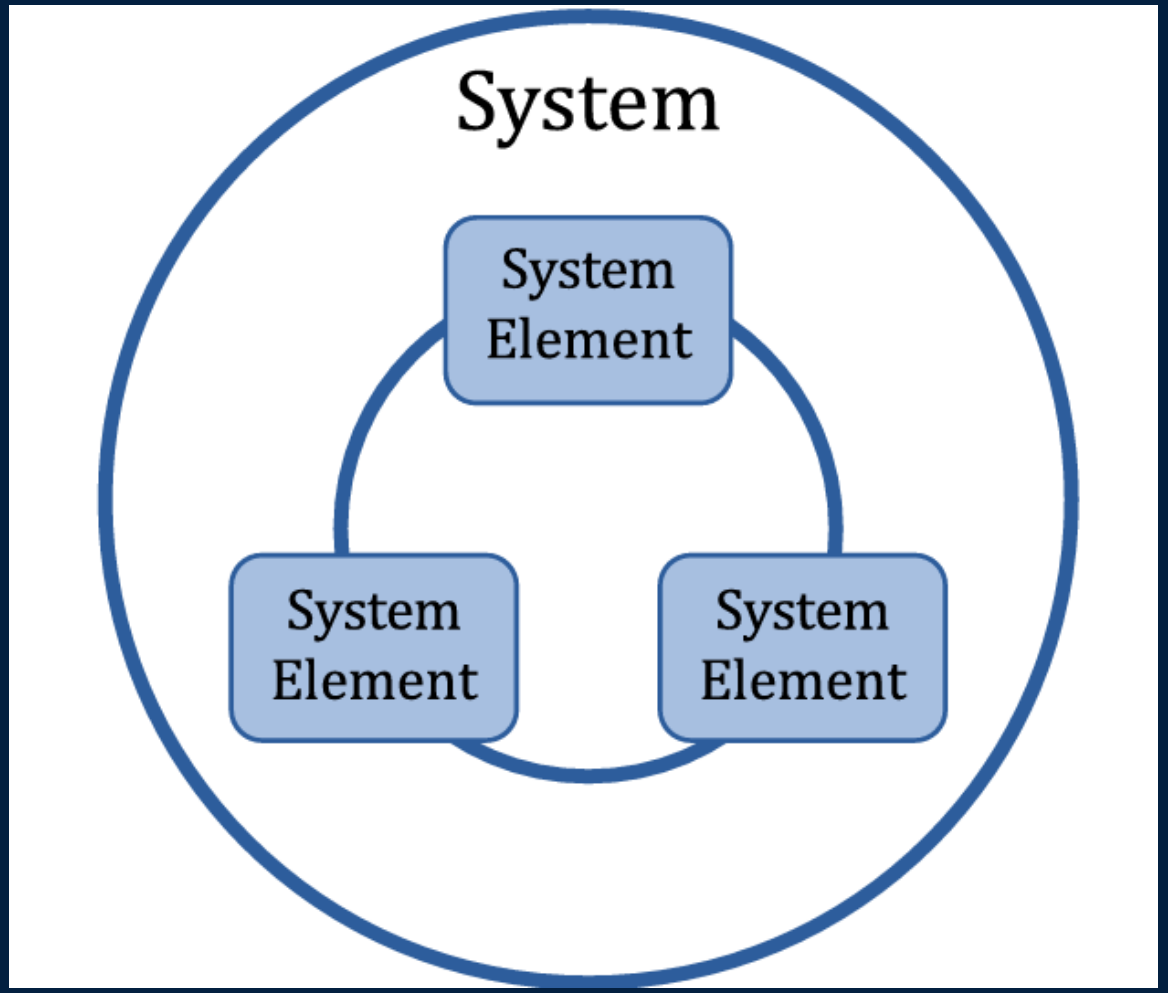
SYSTEM ENGINEERING

Ramin Karim, PhD, Chair Professor



A SYSTEM & INHERENT ELEMENTS

- A system is composed of a set of interacting system elements, each of which can be implemented to fulfil
- its respective specified requirements. System elements may include software elements, hardware elements, services, and utilization and support resources.



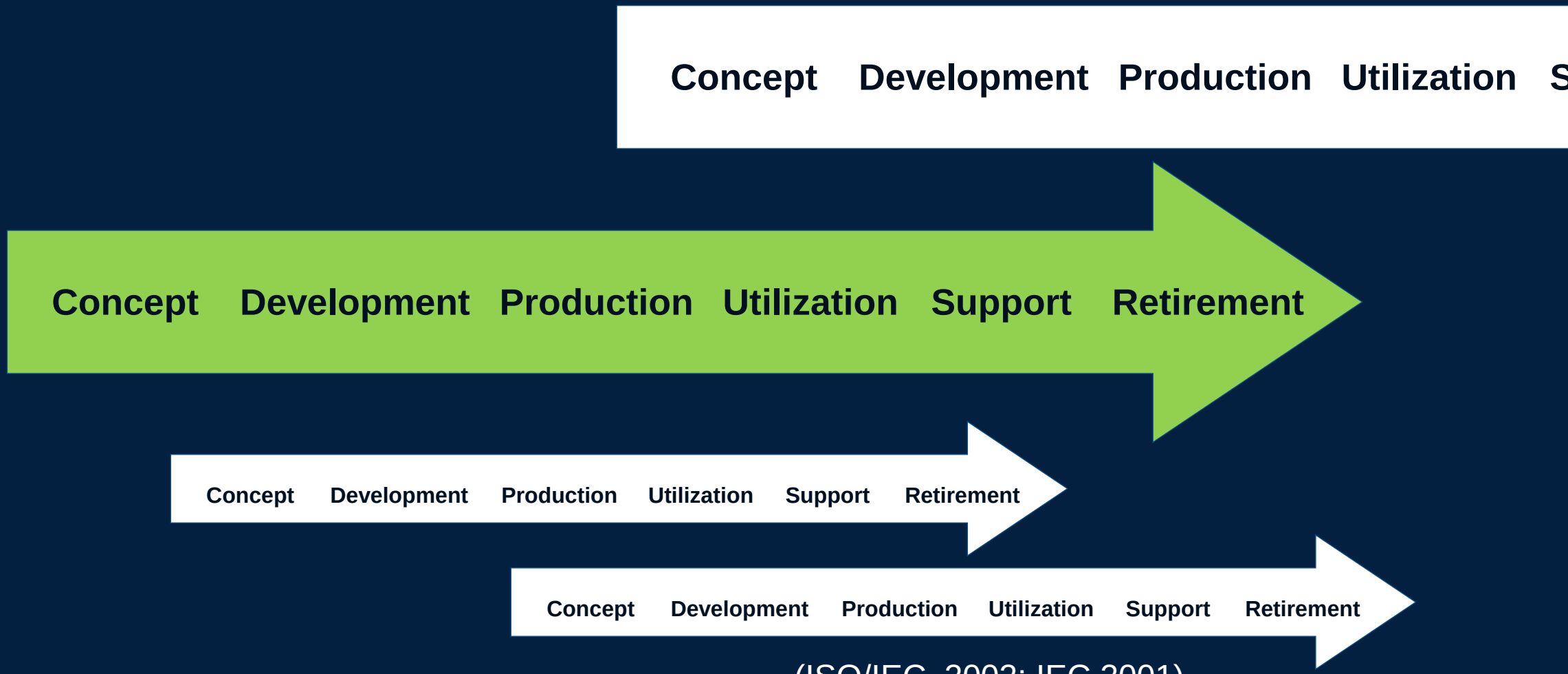
(ISO15288, 2023)

SYSTEM LIFECYCLE PHASE (ISO 15288)

1. Concept – Explore needs and propose solutions.
2. Development – Refine requirements, design and verify.
3. Production – Manufacture and test.
4. Utilization – Operate in intended environment.
5. Support – Maintain and upgrade.
6. Retirement – Safely dispose of or archive.

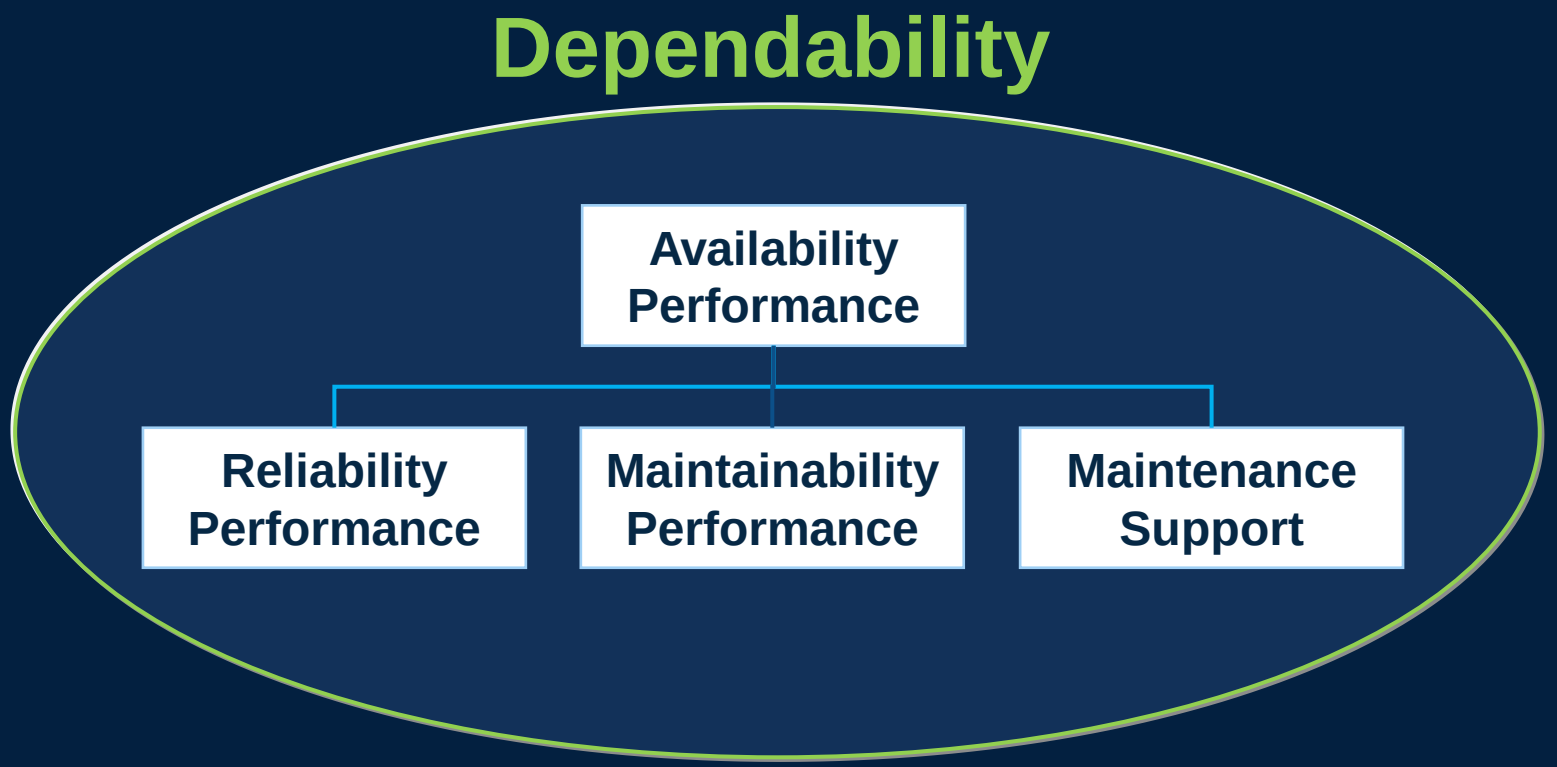
- Vee Model is used to illustrate development and verification across these stages.

LIFECYCLE PERSPECTIVE



(ISO/IEC, 2002; IEC 2001)

THE CONCEPT OF DEPENDABILITY



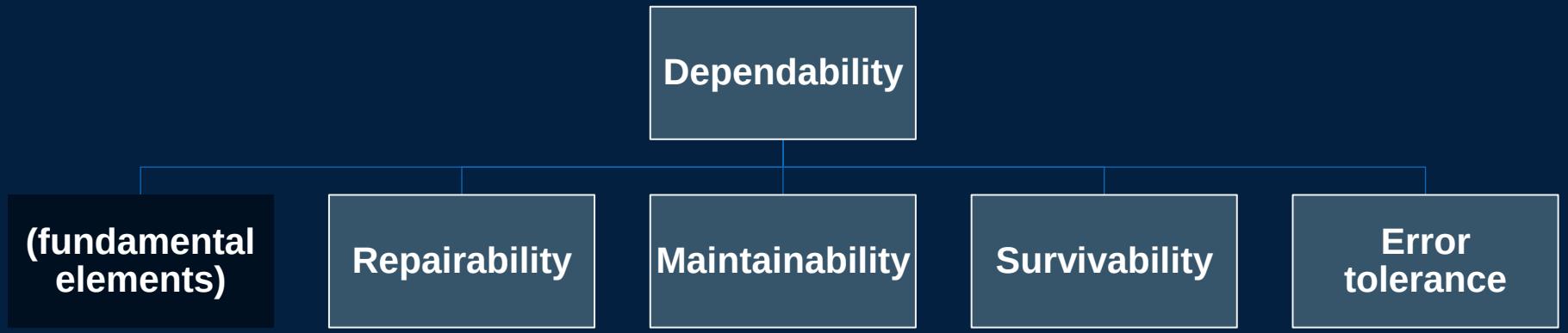
- Dependability of a system implies availability performance and its inherent factors: reliability performance, maintainability performance and maintenance support performance (IEV, 2008)

SOFTWARE ENGINEERING – FUNDAMENTAL ELEMENTS OF DEPENDABILITY

- **Availability** – the ability of the system to deliver services when requested
- **Reliability** - the ability of the system to deliver services as specified
- **Safety** - the ability of the system to operate without catastrophic failure
- **Security** - the ability of the system to protect itself against accidental or deliberate intrusions

(Sommerville, 2007)

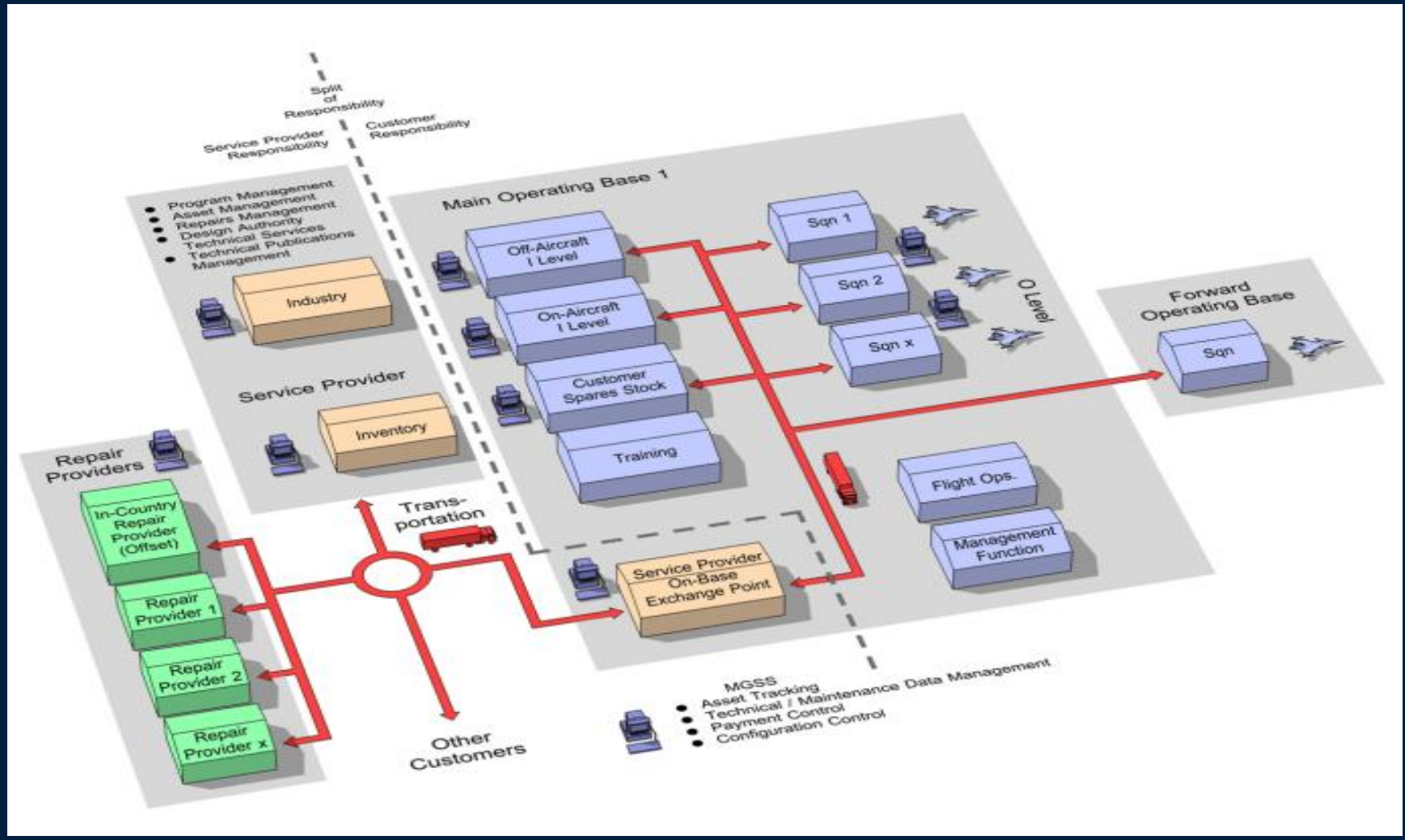
SOFTWARE ENGINEERING – COMPLEMENTARY ELEMENTS OF DEPENDABILITY



- **Repairability** – the ability of the system to be repaired after failure
- **Maintainability** - the ability of the system to adapt to new requirements (modified or reconfigured)
- **Survivability** - the ability of the system to continue to deliver services under attach (partly disabled)
- **Error Tolerance** - the ability of the system to handle unexpected situations (exception handling)

(Sommerville, 2007)

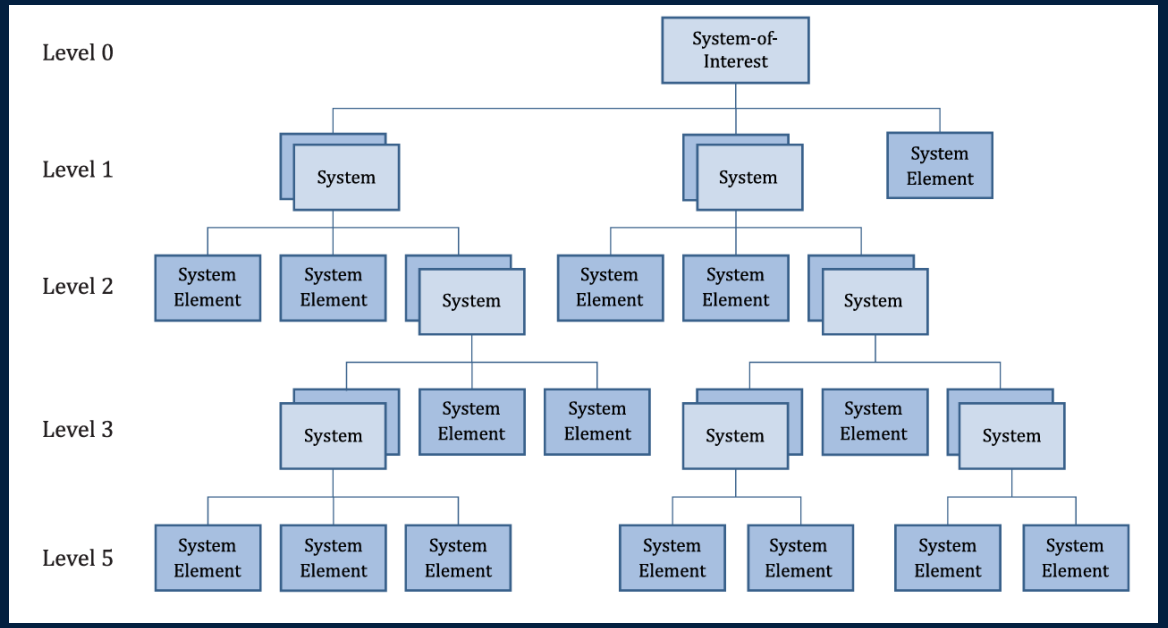
DEPENDABILITY - EXAMPLE



SYSTEM OF INTEREST

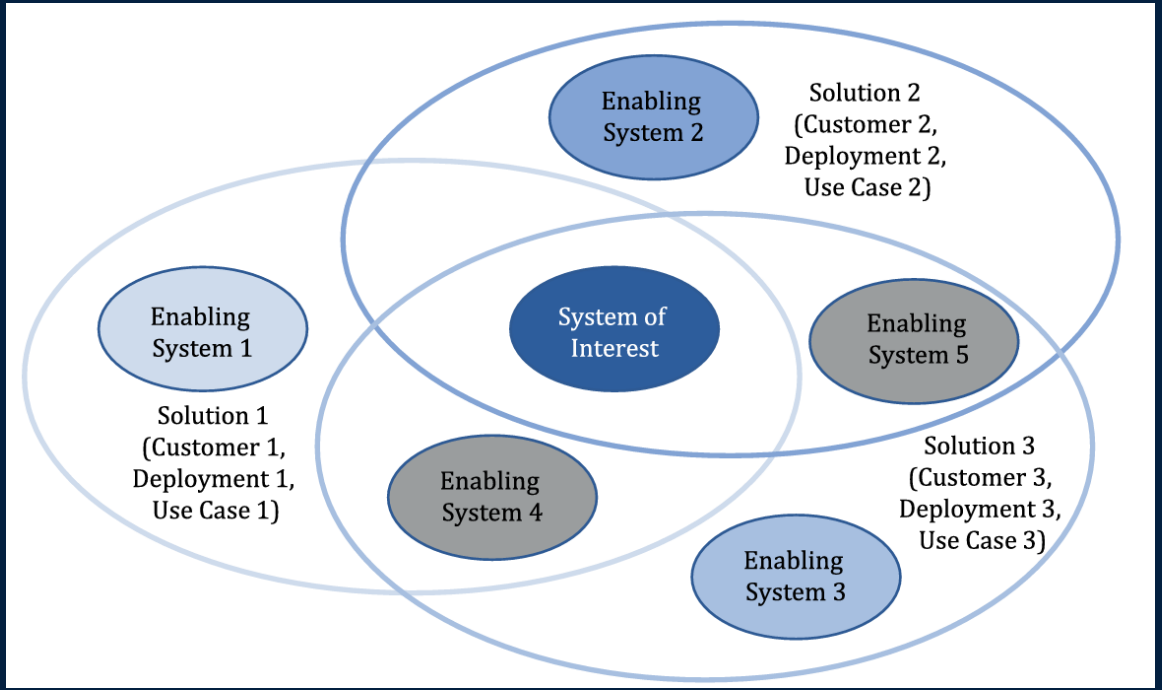
- A system that its lifecycle is under the consideration
- The relationship between system elements can be expressed in many forms, including hierarchies or networks.
- For more complex SoIs, a prospective system element may itself need to be considered as a system (that in turn is comprised of system elements) before a complete set of system elements can be defined with confidence

(ISO15288, 2023)



ENABLING SYSTEMS

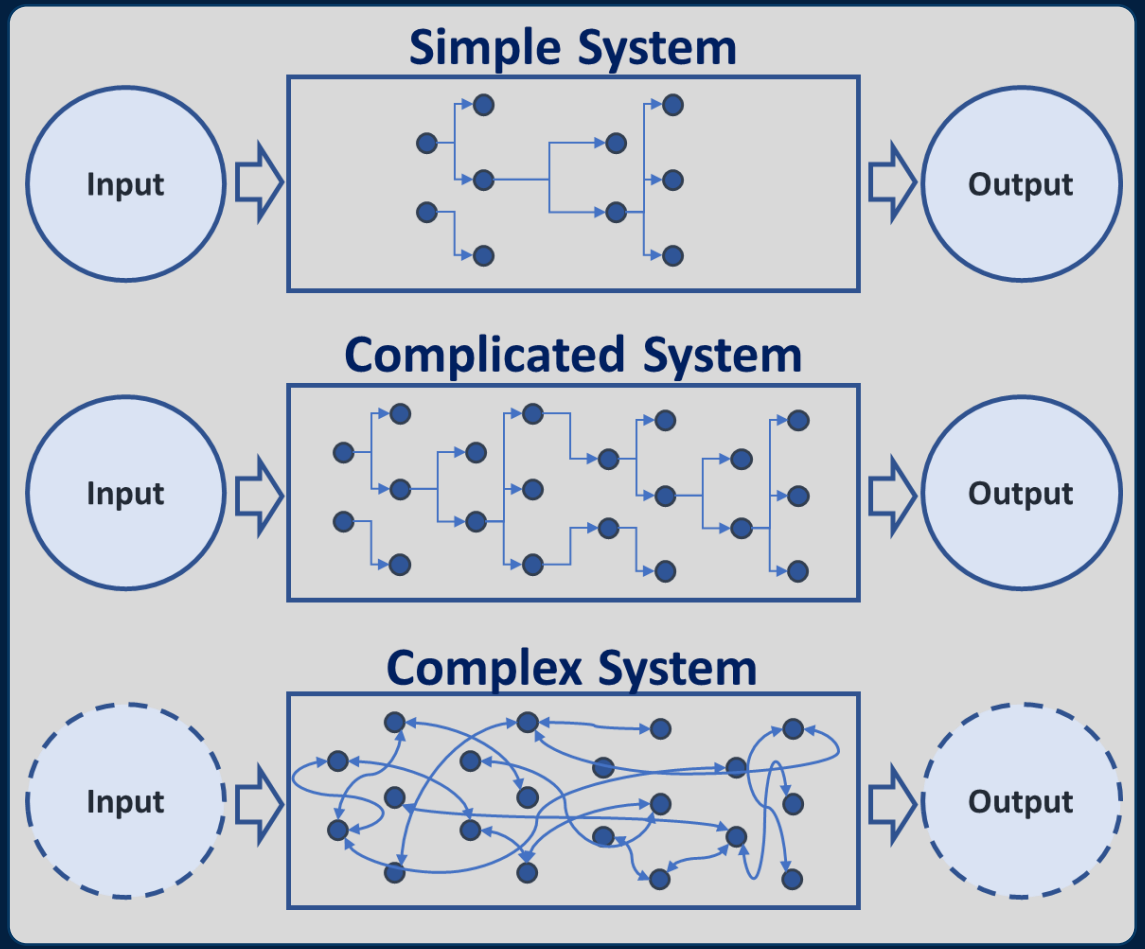
- Any system sharing an interface (data or information, energy, resource, physical) with the SoI during any stage of the SoI's life cycle is an interfacing system and needs to be considered in the system development.
- Humans can be system elements of the SoI (e.g. an operator) or can be interfacing externally to the SoI (e.g. a user requesting information) throughout the SoI's life cycle stages.



(ISO15288, 2023)

A SYSTEM PERSPECTIVE

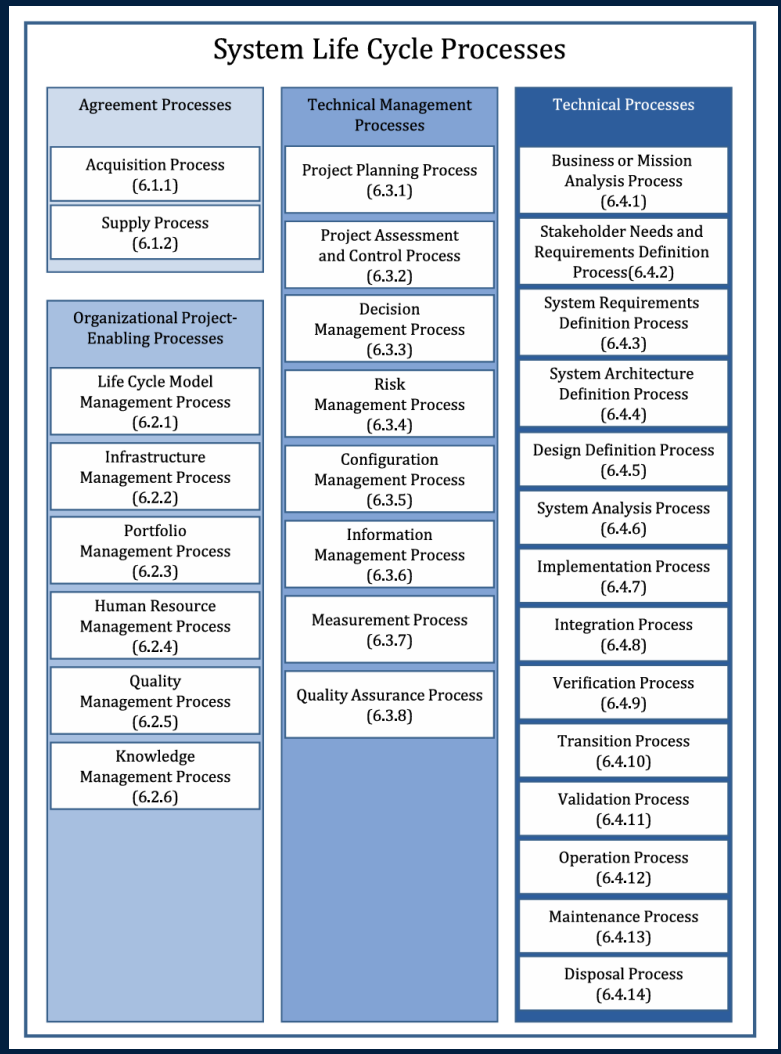
- *Simple, Complicated & Complex* system-of-systems



SYSTEMS AND SOFTWARE ENGINEERING — SYSTEM LIFE CYCLE (ISO/IEC 15288)

- ISO15288 establishes a common framework of process descriptions for describing the life cycle of systems created by humans, defining a set of processes and associated terminology from an engineering viewpoint.
  - These processes can be applied to systems of interest, their system elements, and to systems of systems. Selected sets of these processes can be applied throughout the stages of a system's life cycle.
  - This is accomplished through the involvement of stakeholders, with the ultimate goal of achieving customer satisfaction.
- ISO15288 defines a set of processes to facilitate system development and information exchange among acquirers, suppliers, and other stakeholders in the life cycle of a system.

- ISO15288 groups the activities that can be performed during the life cycle of a system into four process groups:
  a) agreement processes
  b) organizational project-enabling processes
  c) technical management processes
  d) technical processes

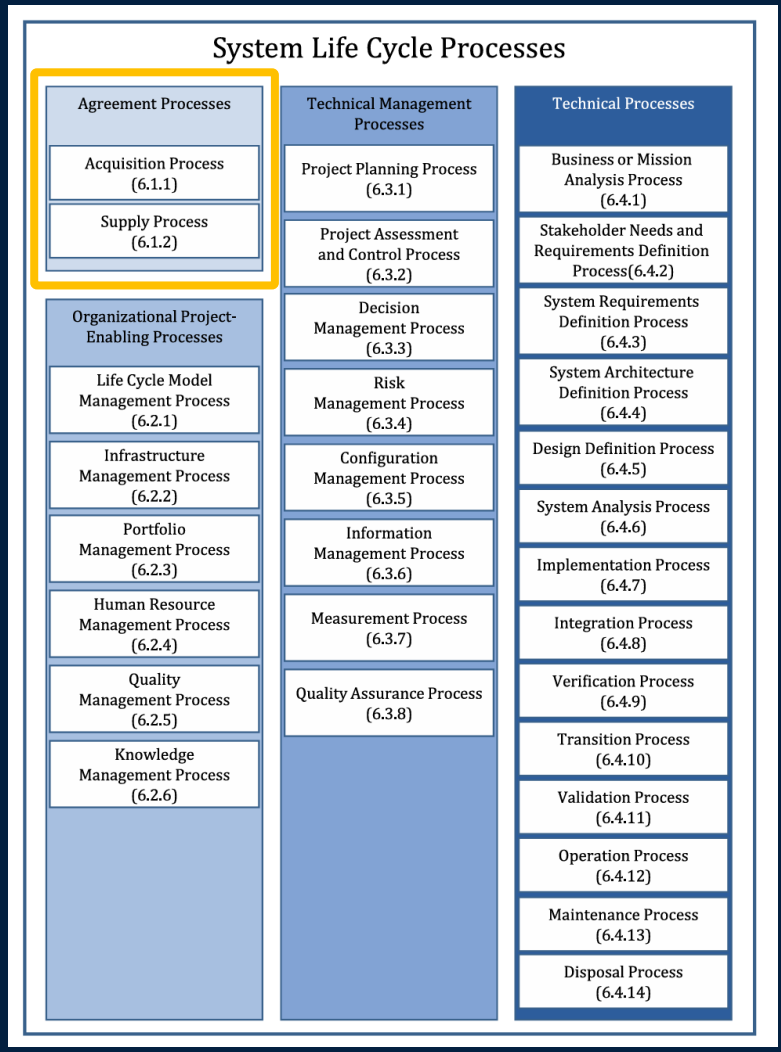


(ISO15288, 2023)

SE - AGREEMENT PROCESSES

- The agreement processes consist of the following:
  - a) acquisition process – used by organizations for acquiring products or services
  - b) supply process – used by organizations for supplying products or services

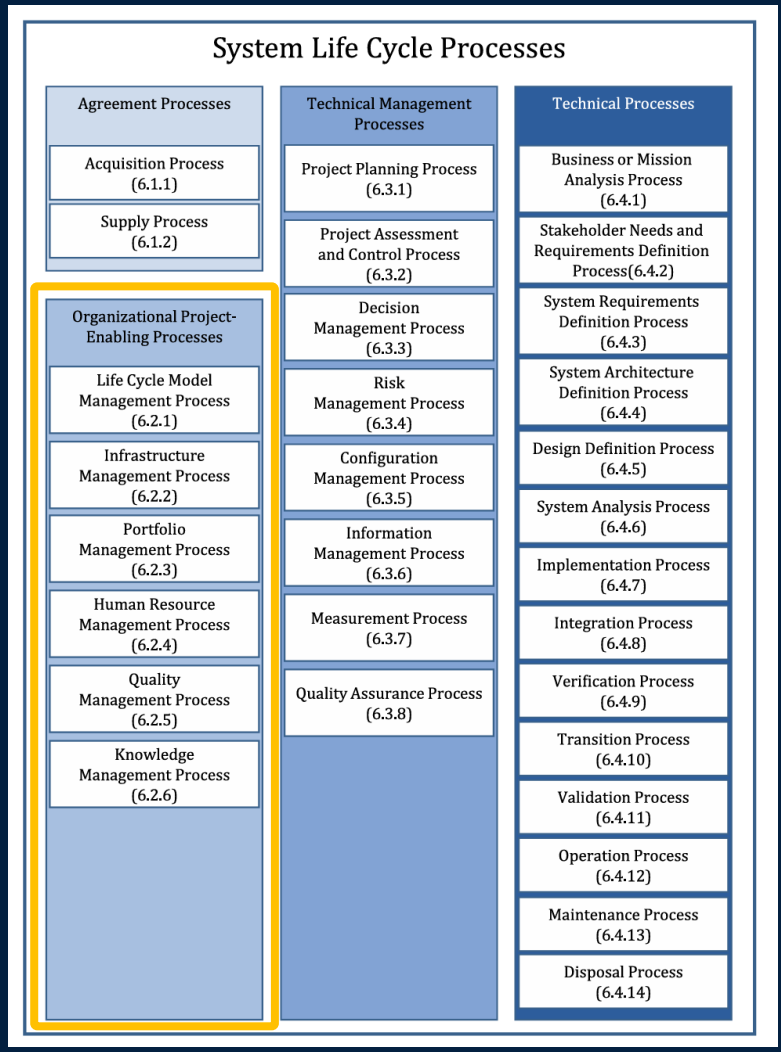
- These processes define the activities necessary to establish an agreement between two organizations
  - If the acquisition process is invoked, it provides the means for interacting with a supplier.



(ISO15288, 2023)

SE - ORGANISATIONAL PROJECT-ENABLING PROCESSES

- The organizational project-enabling processes are concerned with providing the resources needed to enable the project to meet the needs and expectations of the organization's stakeholders
  a) life cycle model management process
  b) infrastructure management process
  c) portfolio management process
  d) human resource management process
  e) quality management process
  f) knowledge management process

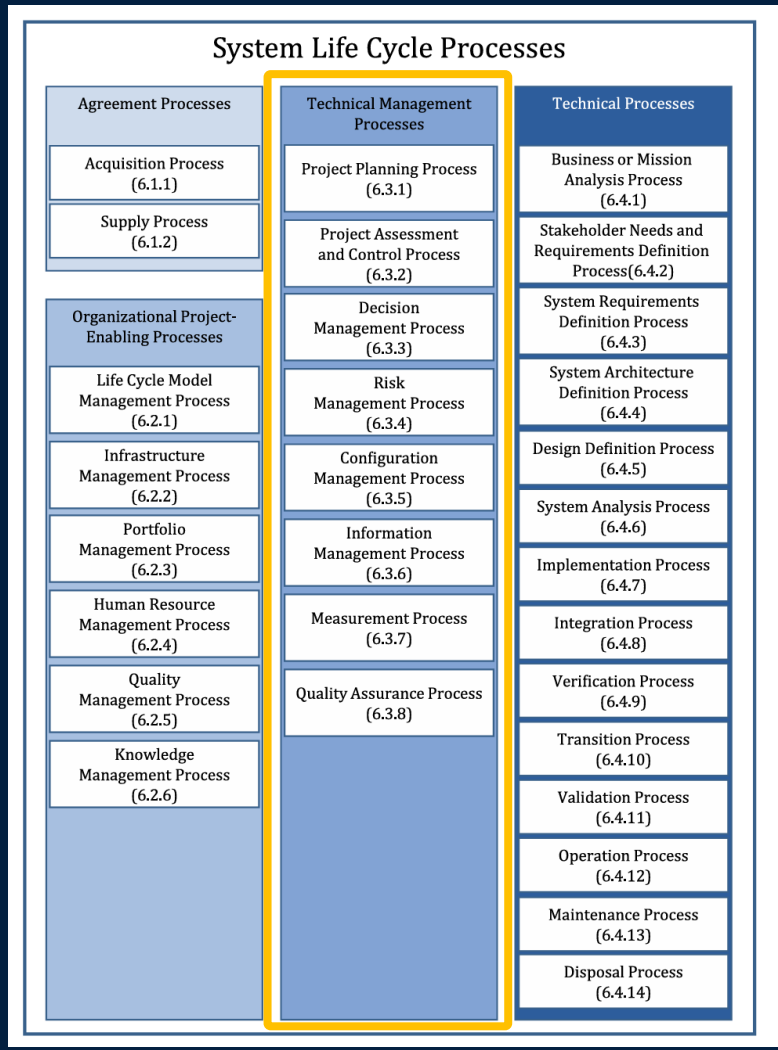


(ISO15288, 2023)

SE - TECHNICAL MANAGEMENT PROCESSES

- The technical management processes relate to the technical effort of projects, in particular to planning in terms of cost, timescales, and achievements; to the checking of actions to help ensure that they comply with plans and performance criteria; and to the identification and selection of corrective actions that recover shortfalls in progress and achievement.
  a) project planning process;
  b) project assessment and control process;
  c) decision management process;
  d) risk management process;
  e) configuration management process;
  f) information management process;
  g) measurement process;
  h) quality assurance process.

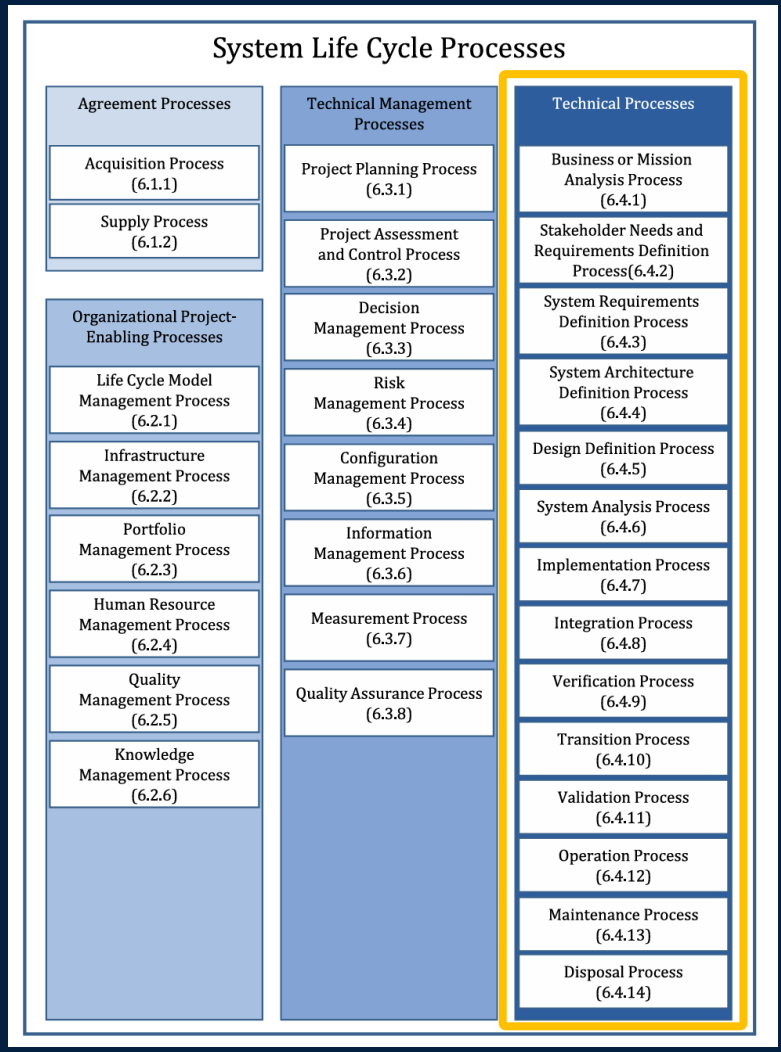
(ISO15288, 2023)



SE - TECHNICAL PROCESSES

- The technical processes are concerned with technical actions throughout the life cycle. Technical processes transform the needs of stakeholders into products or services.
  - a) business or mission analysis process;
  - b) stakeholder needs and requirements definition process;
  - c) c) system requirements definition process;
  - d) d) system architecture definition process;
  - e) e) design definition process;
  - f) f) system analysis process;
  - g) g) implementation process;
  - h) h) integration process;
  - i) i) verification process;
  - j) j) transition process;
  - k) k) validation process;
  - l) l) operation process;
  - m) m) maintenance process;
  - n) n) disposal process.

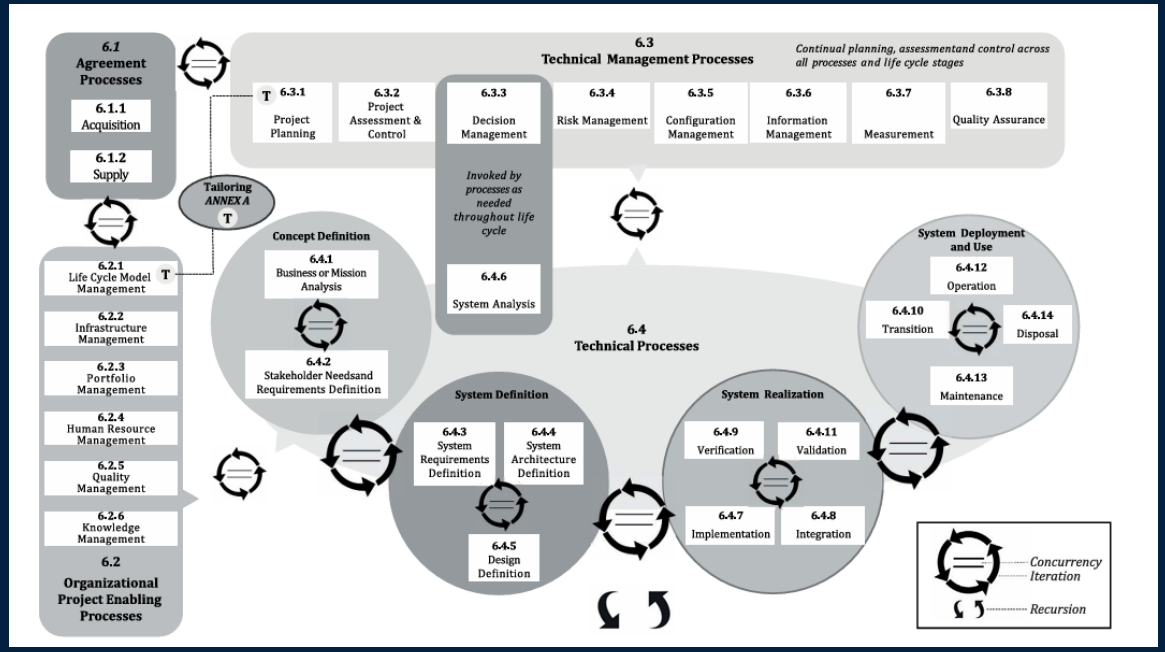
(ISO15288, 2023)



INTERRELATIONSSHIPS BETWEEN PROCESSES

- The processes are often applied using an MBSE approach, which uses a set of models to implement the processes and achieve the expected outcomes. See Annex D for information regarding MBSE.

- (MBSE = Model-Based Systems Engineering)



(ISO15288, 2023)

METHODOLOGY, METHOD, MODEL

- **Methodology**, is about finding appropriate procedure to solve problem, e.g. Participatory Research
- **Method**, is a research tool used to solve a specific problem, e.g. Case Study
- **Model** is a representation of something

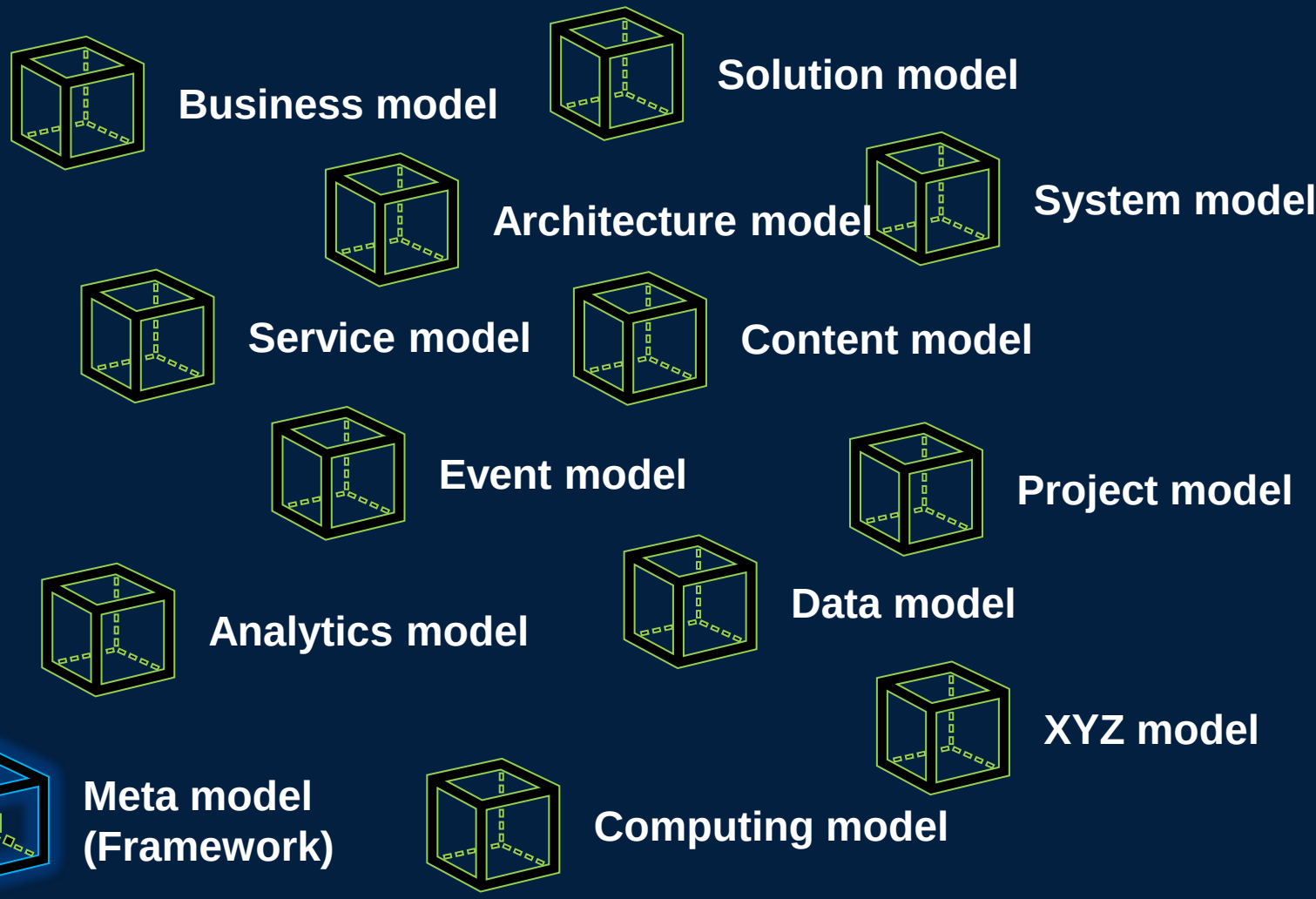
In research one uses **methodology**

to select **method**

to develop a **model**

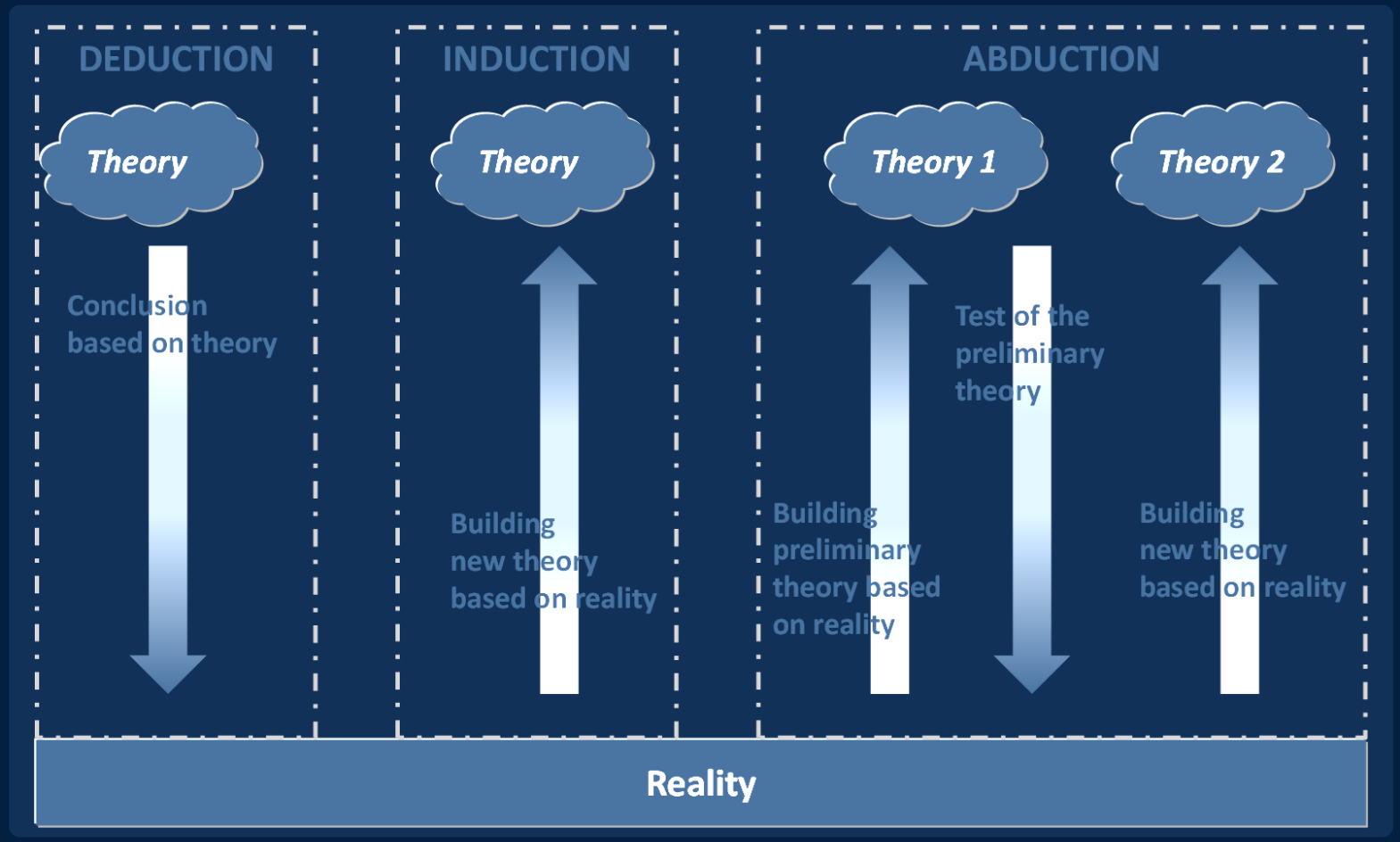
IT IS ALL ABOUT MODELLING

- A MODEL is a representation of something!
  - “Modelling is the designing of software applications before coding” (UML,2020)
- Notations
  - Formal languages
  - Example: Unified Modelling Language (UML), SysML, RUP,
    - https://www.uml.org/



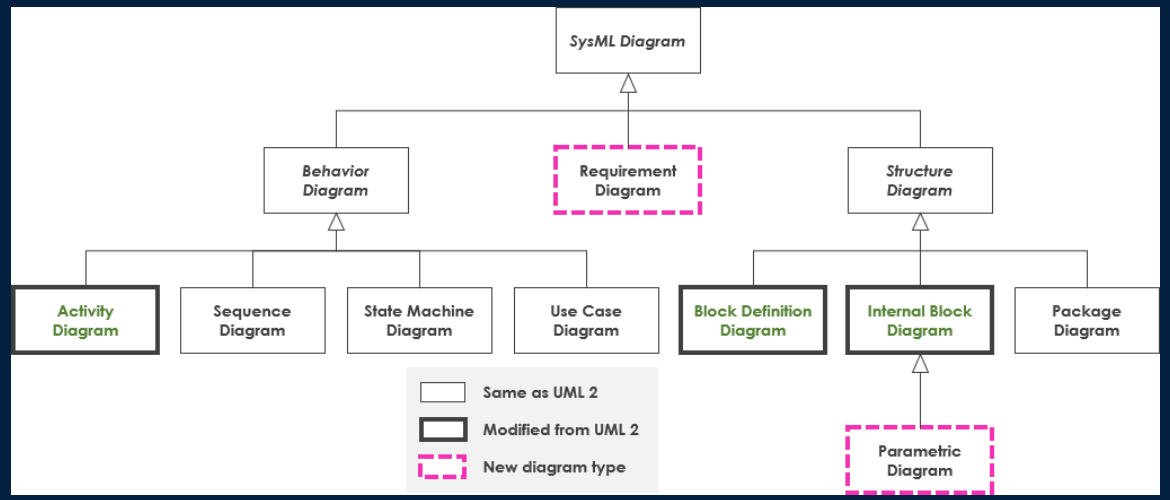
AI & REASONING

- Data-driven
- Model-driven
- Information-driven
- Hybrid



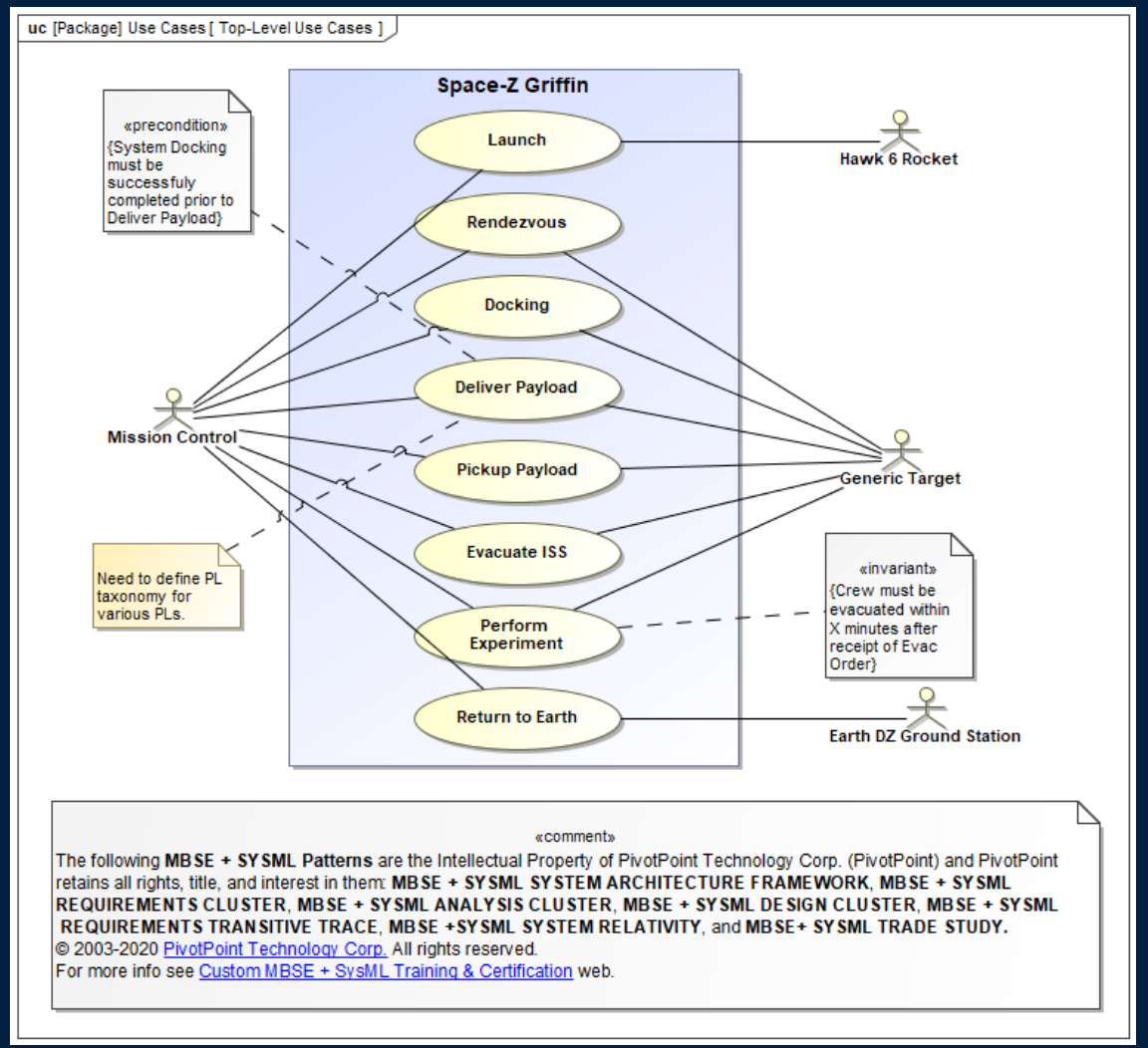
VARIOUS TYPES OF SYSML DIAGRAMS

1. **Structure Diagrams**: Used to represent the physical or logical architecture of a system. Key types include:
    1. **Block Definition Diagrams (BDDs)**: Define the system's structure in terms of blocks and their relationships.
    2. **Internal Block Diagrams (IBDs)**: Show the internal structure of a block, including its parts and connections.
    3. **Package Diagrams**: Organize model elements into packages.
2. **Requirement Diagrams**: Focus on defining and managing system requirements. This includes:
    1. **Requirement Diagrams**: Capture and manage system requirements.
3. **Behavior Diagrams**: Represent the dynamic behavior of a system, including activities, states, and interactions. Key types include:
    1. **Activity Diagrams**: Model the flow of control and data within a system.
    2. **Sequence Diagrams**: Show the sequence of interactions between system components.
    3. **State Machine Diagrams**: Represent the states and transitions of a system or component.
    4. **Use Case Diagrams**: Capture the functional requirements of a system.



MBSE TOOLS

- SysML, or Systems Modeling Language, is a general-purpose modeling language specifically designed for systems engineering applications.
  - It supports the specification, analysis, design, verification, and validation of a broad range of systems and systems-of-systems
  - SysML was developed as an extension of a subset of the Unified Modeling Language (UML) using UML's profile mechanism.



UML

- UML 2.0 defines thirteen types of diagrams, divided into three categories: Six diagram types represent static application structure; three represent general types of behavior; and four represent different aspects of interactions:

  - Structure Diagrams include the Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram.

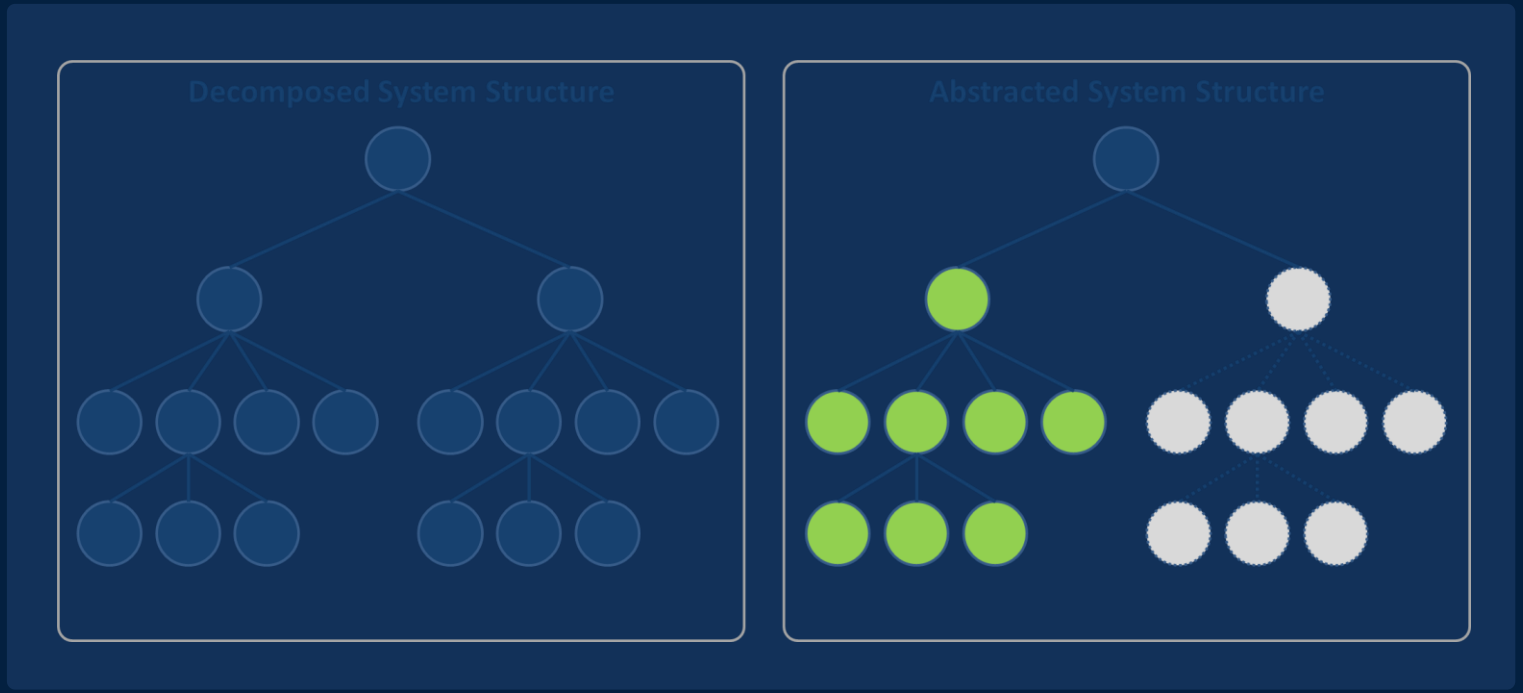
  - Behavior Diagrams include the Use Case Diagram (used by some methodologies during requirements gathering); Activity Diagram, and State Machine Diagram.

  - Interaction Diagrams, all derived from the more general Behavior Diagram, include the Sequence Diagram, Communication Diagram, Timing Diagram, and Interaction Overview Diagram.

(uml.org, 2020)

AI & SYSTEM THINKING & COMPUTER SCIENCE

- Decomposition & abstraction affects AI
  - Model selection
  - Data relevance
  - Information logistics
  - Model precision
  - Explainability
  - …
- Examples
  - Fleet Management
    - Fleet2Individual2Fleet
  - Function Management
    - PHM2PFM



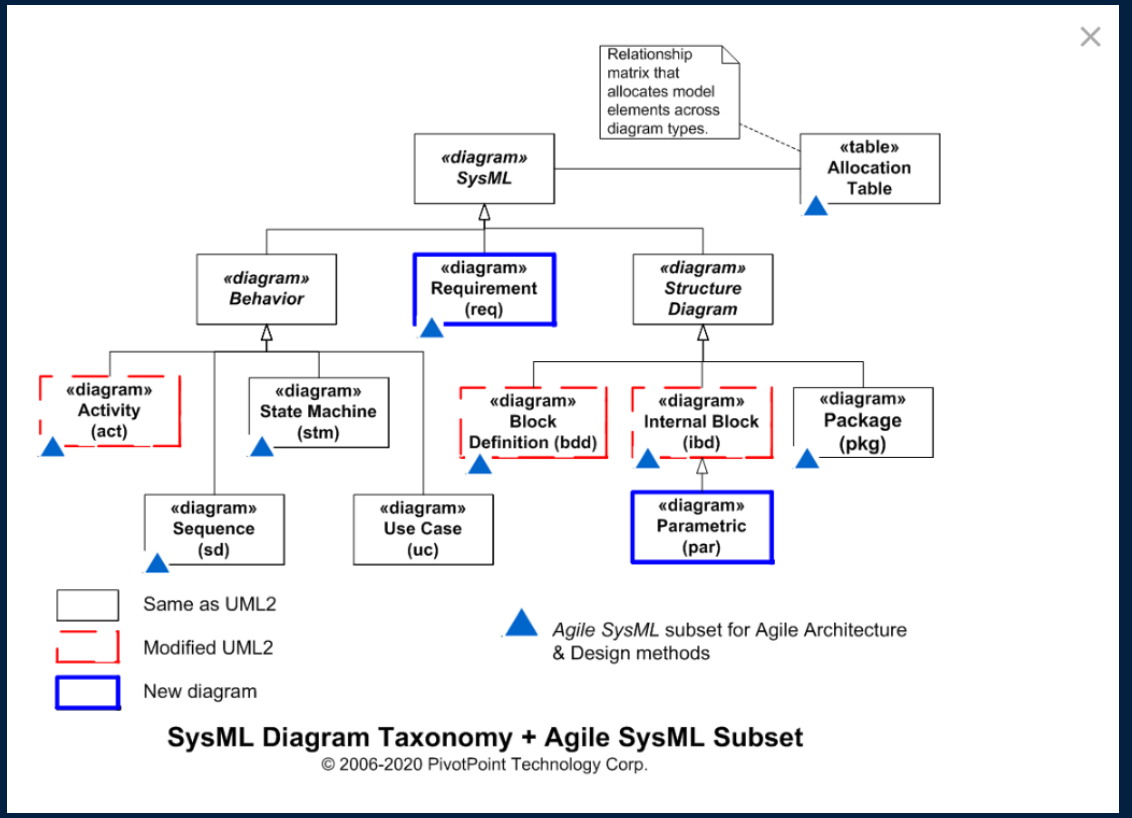
SYSTEM ENGINEERING

- **Systems Modeling Language (SysML)**: SysML is a general-purpose architecture modeling language for Systems Engineering applications.

- SysML supports the specification, analysis, design, verification and validation of a broad range of **systems** and **systems-of-systems**. These systems may include hardware, software, information, processes, personnel, and facilities.

- SysML is a dialect of UML 2, and is defined as a UML 2 Profile. (A UML Profile is a UML dialect that customizes the language via three mechanisms: Stereotypes, Tagged Values, and Constraints.)

(sysml.org, 2020)



INCOSE

- **INCOSE** stands for the **International Council on Systems Engineering**. It is a **not-for-profit professional organization** founded in 1990, dedicated to advancing the discipline and practice of **systems engineering (SE)**.
- **Mission**: To address complex system challenges through systems engineering.
- **Members**: Engineers, scientists, educators, managers, and students from government, industry, and academia.
- **Global Reach**: Chapters all over the world, including in the U.S., Europe, Asia, Africa, and Australia.
- **Core Activities**:
  - Publishing **standards**, **handbooks**, and **guides**
  - Organizing **conferences**, **workshops**, and **working groups**.
  - Offering **certifications** such as:
    - **ASEP** – Associate Systems Engineering Professional
    - **CSEP** – Certified Systems Engineering Professional
    - **ESEP** – Expert Systems Engineering Professional

“FIRST, YOU MUST UNLEARN WHAT YOU HAVE LEARNED!”

(Jedi Master Yoda)

Thank You for Your Attention!