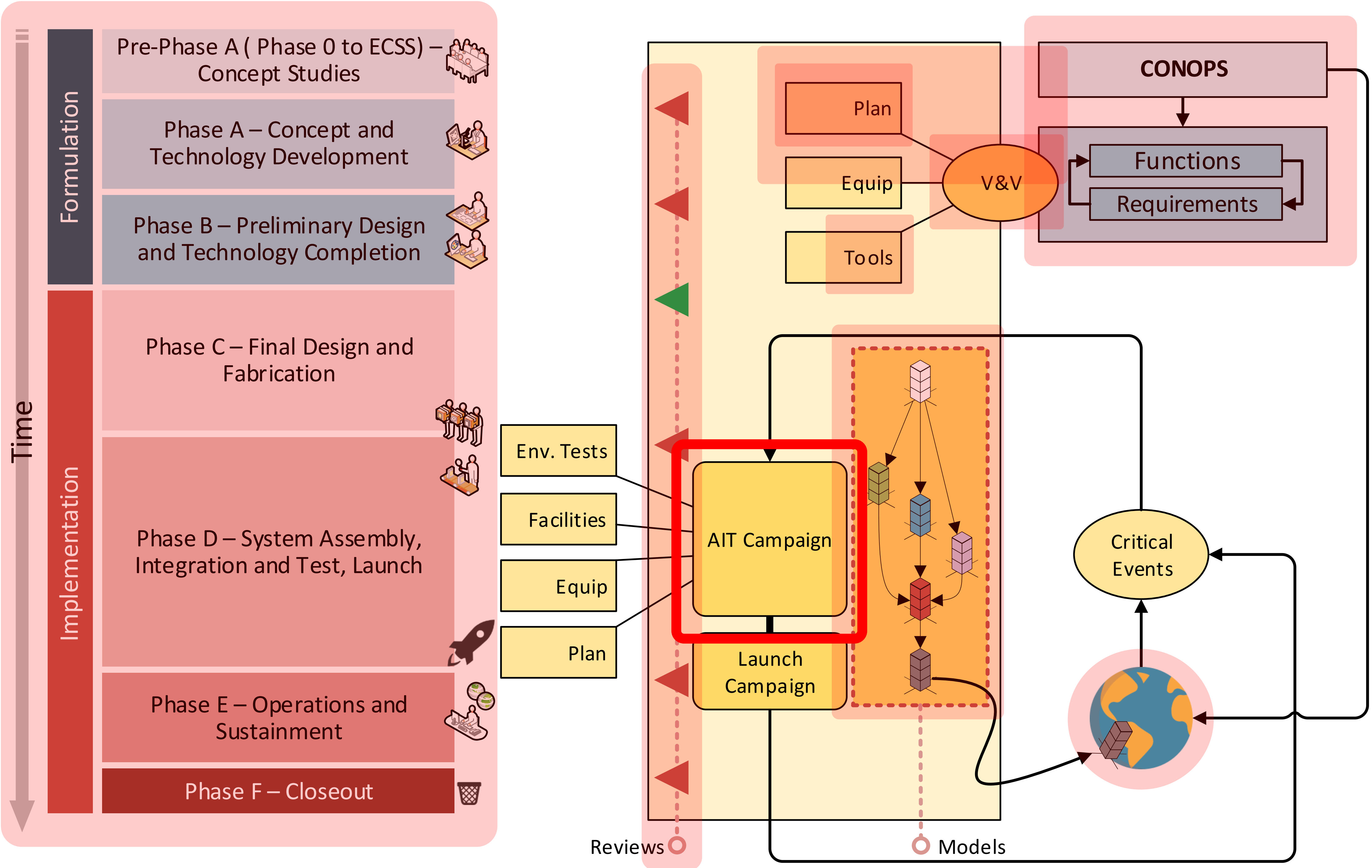
AIT Process

[SIS-08][LEC-008]

Date	SES	In Class	Deliverables
Aug, 1	01	[LEC-000] Course Introduction [LEC-001] SE Review	-
Aug, 8	02	[LEC-002] Global Verification Process	[PRD-001] ~~System Description & Architecture~~
Aug, 15	03	[LEC-003] Tool and Processes to Verification	[PRD-002] ~~System DSM~~ Product Tree
Aug, 22	04	[LEC-004] Life Cycle, Reviews & Baselines	[PRD-003] Revised Requirements
Aug, 29	05	[LEC-005] Model Philosophy	[PRD-004] Verifications per Requirement through the Life Cycle
Sep, 5	06	[LEC-006] Preparing to test Campaigns	[PRD-005] Models
Sep, 12	07	[LEC-007] Planning V&V	[PRD-006] Test Articles, Procedures & VCD
Sep, 19	08	[TST-001] V&V Conceptual Questions [PRD-007] DRAFT V&V Plan Presentation	[PRD-007] DRAFT V&V Plan (DVM)
Sep, 26	Week off		
Oct, 03	09	[LEC-008] AIT Process	[PRD-008] End to End Test Articles
Oct, 10	10	[LEC-009] Critical Events & Environmental Tests	[PRD-009] AIT Activities through the Life Cycle
Oct, 17	11	[LEC-010] Testing Facilities	[PRD-010] Vehicle and On-Orbit Testing
Oct, 24	12	[LEC-011] Planning AIT	[PRD-011] Facilities
Oct, 31	13	[LEC-012] GSEs [LEC-013] SCOE/OCOE	[PRD-012] AIT Flows & Activity Log
Nov, 07	14	[LEC-014] Launching Campaign	[PRD-013] GSEs
Nov, 14	15	[LEC-015] Trends / MBSE / Industry 4.0	[PRD-014] AIT Task Sheets [PRD-015] Vehicle Integration & Launching Plan
Nov, 21	16	[TST-002] AIT Conceptual Questions [PRD-016] V&V & AIT Plans Presentation	[PRD-016] V&V & AIT Plans
Nov, 28 Dez, 05	EXAM: Design of an AIT Facility to ITA's SmallSat Projects		

CONOPS



Time
Formulation
Implementation
Pre-Phase A (Phase 0 to ECSS) – Concept Studies
Phase A – Concept and Technology Development
Phase B – Preliminary Design and Technology Completion
Phase C – Final Design and Fabrication
Phase D – System Assembly, Integration and Test, Launch
Phase E – Operations and Sustainment
Phase F – Closeout
Env. Tests
Facilities
Equip
Plan
Plan
Equip
Tools
V&V
CONOPS
Functions
Requirements
AIT Campaign
Launch Campaign
Critical Events
Reviews
Models

SUMMARY

- AIT Definition
- AIT Process
- DSM (the return)

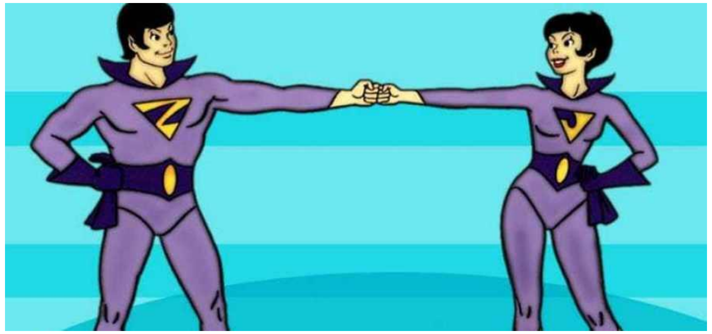
AIT DEFINITION

What is AIT?

- The Assembly, Integration and Tests activities includes:
  - The implementation of a **group of procedures**;
  - The execution of a **sequence of events**
  - The purpose is to obtain the **operational working trust**, thus, guarantee that all project parameters and specified performances are meet.

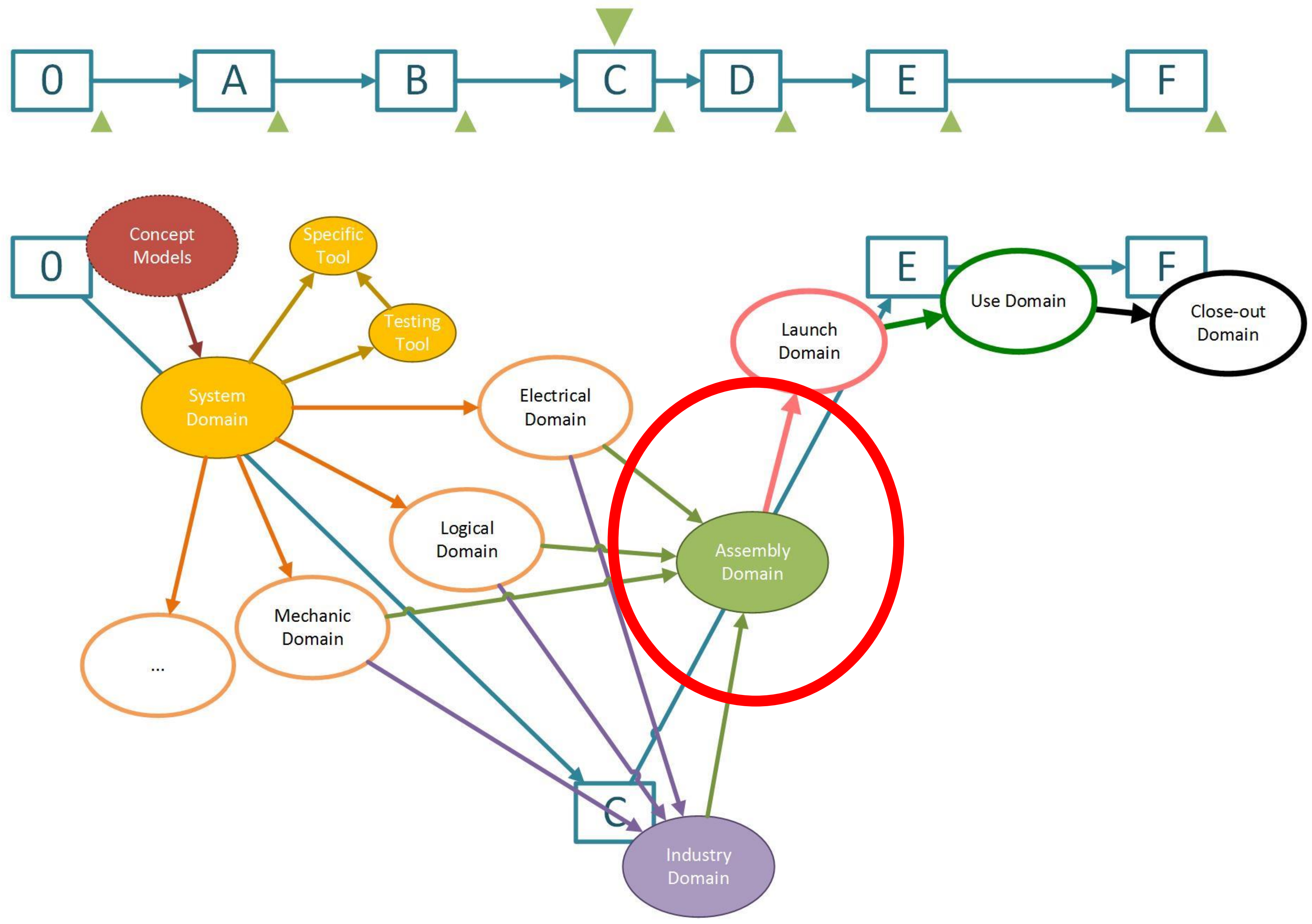
- **Assembly**: group of **mechanical operations** executed to position, fasten and **physically interconnect** each one of the components of a system.

- **Integration**: **functional interconnection and verification** among the system's components.



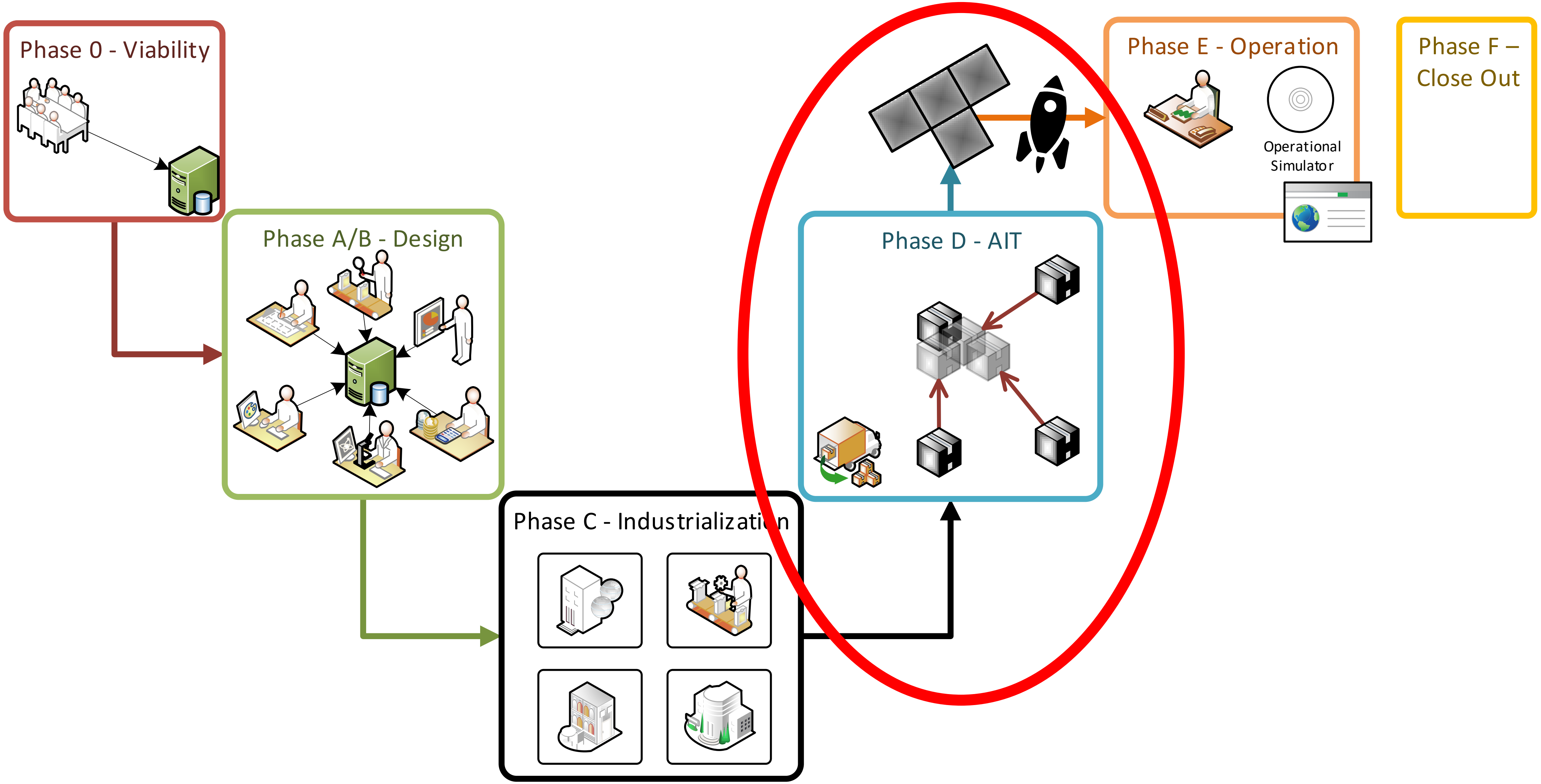
- **Test**: all **test sequences** to verify the requirements and the survivability of the system in the use environmental.

Domain

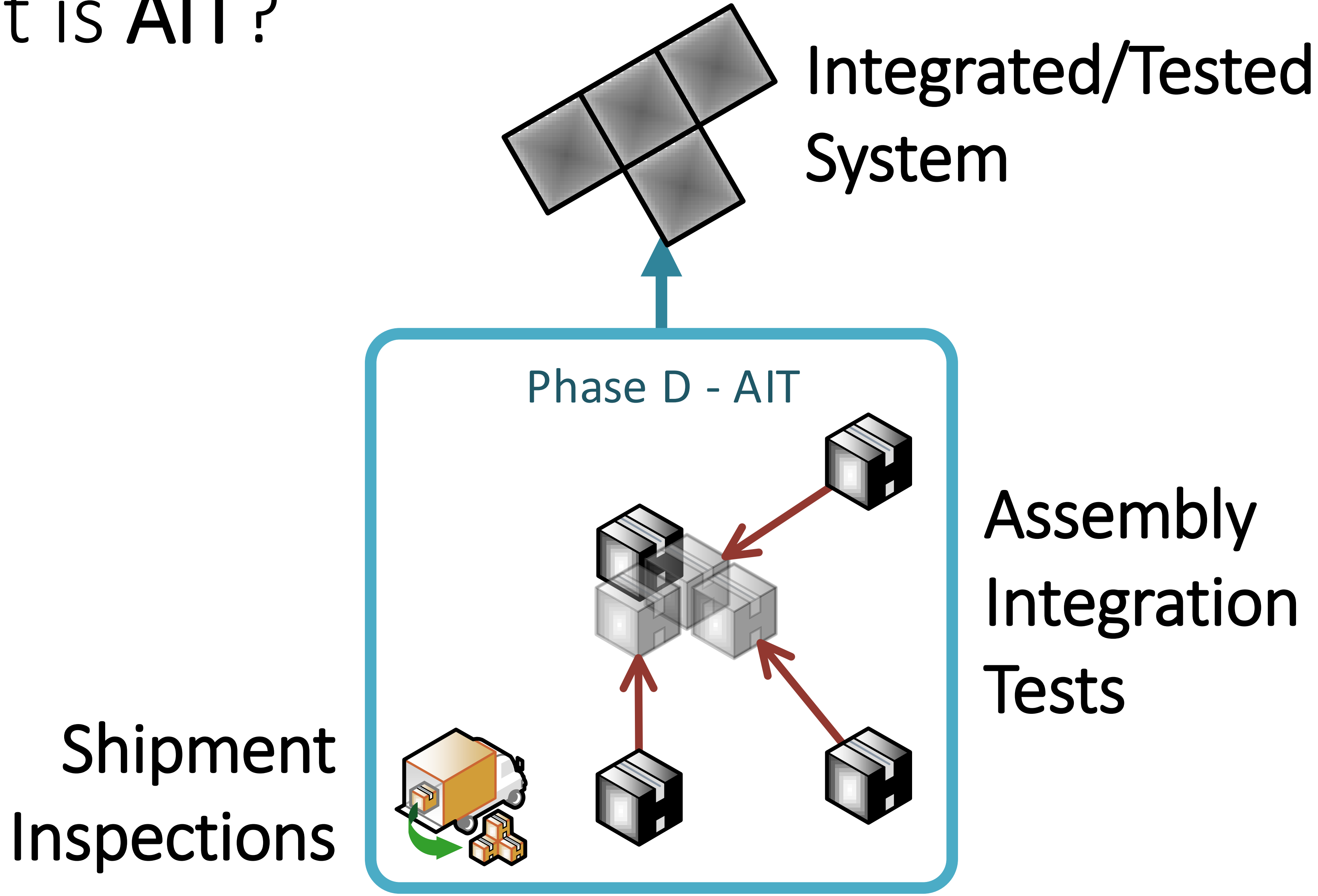


0
A
B
C
D
E
F
0
Concept Models
Specific Tool
Testing Tool
System Domain
Electrical Domain
Logical Domain
Mechanic Domain
...
Assembly Domain
Launch Domain
E
Use Domain
F
Close-out Domain
C
Industry Domain

Into the System Engineering Process



What is **AIT**?


Integrated/Tested
System
Phase D - AIT
Assembly
Integration
Tests
Shipment
Inspections

"Product integration is a key activity of the systems engineer. Product integration is the *engineering of the subsystem* ***interactions*** *and their* ***interactions*** *with the system environments* (both natural and induced)."

AIT PROCESS

AIT THROUGH LIFECYCLE

- Integration **begins with concept development**, ensuring that the system concept has all necessary functions and major elements and that the induced and natural environment domains in which the system is expected to operate are all identified.
- Integration **continues during requirements development,** ensuring that all system and environmental requirements are compatible, and that the system has a proper balance of functional utility to produce a robust and efficient system.
- Interfaces are defined in this phase and are the pathway of system interactions. Interfaces include mechanical (i.e., structure, loads), fluids, thermal, electrical, data, logical (i.e., algorithms and software), and human. These interfaces may include support for assembly, maintenance, and testing functions in addition to the system main performance functions. The interactions that occur through all of these interfaces can be subtle and complex, leading to both intended and unintended consequences. All of these interactions need to be engineered to produce an elegant and balanced system.

AIT THROUGH LIFECYCLE

- Integration **during the design phase continues** the engineering of these interactions and requires constant analysis and management of the subsystem functions and the subsystem interactions between themselves and with their environments.

- Analysis of the system interactions and managing the balance of the system is the central function of the systems engineer during the design process. The system needs to create and maintain a balance between the subsystems, optimizing the system performance over any one subsystem to achieve an elegant and efficient design. The design phase often involves development testing at the component, assembly, or system level. This is a key source of data on system interactions, and the developmental test program should be structured to include subsystem interactions, human-in-the-loop evaluations, and environmental interaction test data as appropriate.

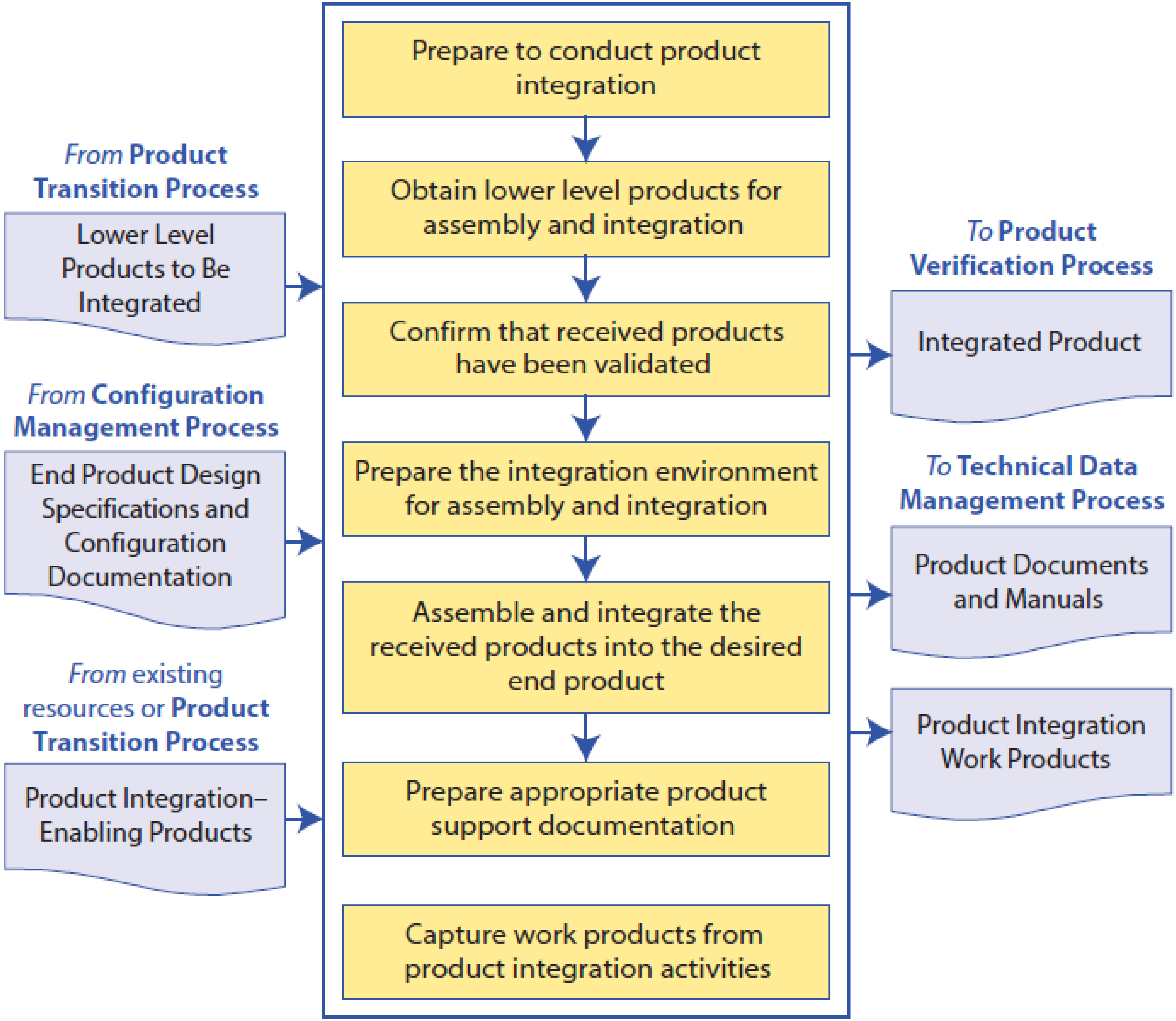
AIT THROUGH LIFECYCLE

- Integration **continues during the operations phase**, bringing together the system hardware, software, and human operators to perform the mission. The interactions between these three integrated natures of the system need to be managed throughout development and into operations for mission success.

- The systems engineer, program manager, and the operations team (including the flight crew from crewed missions) need to work together to perform this management.
- The systems engineer is not only cognizant of these operations team interactions but is also involved in the design responses and updates to changes in mission parameters and unintended consequences (through fault management).

PROCESS INPUTS

- **Lower-level products to be integrated:** These are the products developed in the previous lower-level tier in the product hierarchy. These products will be integrated / assembled to generate the product for this product layer.
- **End product design specifications and configuration documentation:** These are the specifications, Interface Control Documents (ICDs), drawings, integration plan, procedures or other documentation or models needed to perform the integration including documentation for each of the lower-level products to be integrated.
- **Product integration-enabling products:** These would include any enabling products, such as holding fixtures, necessary to successfully integrate the lower-level products to create the end product for this product layer.


From Product Transition Process
Lower Level Products to Be Integrated
From Configuration Management Process
End Product Design Specifications and Configuration Documentation
From existing resources or Product Transition Process
Product Integration–Enabling Products
Prepare to conduct product integration
Obtain lower level products for assembly and integration
Confirm that received products have been validated
Prepare the integration environment for assembly and integration
Assemble and integrate the received products into the desired end product
Prepare appropriate product support documentation
Capture work products from product integration activities
To Product Verification Process
Integrated Product
To Technical Data Management Process
Product Documents and Manuals
Product Integration Work Products

PROCESS OUTPUTS

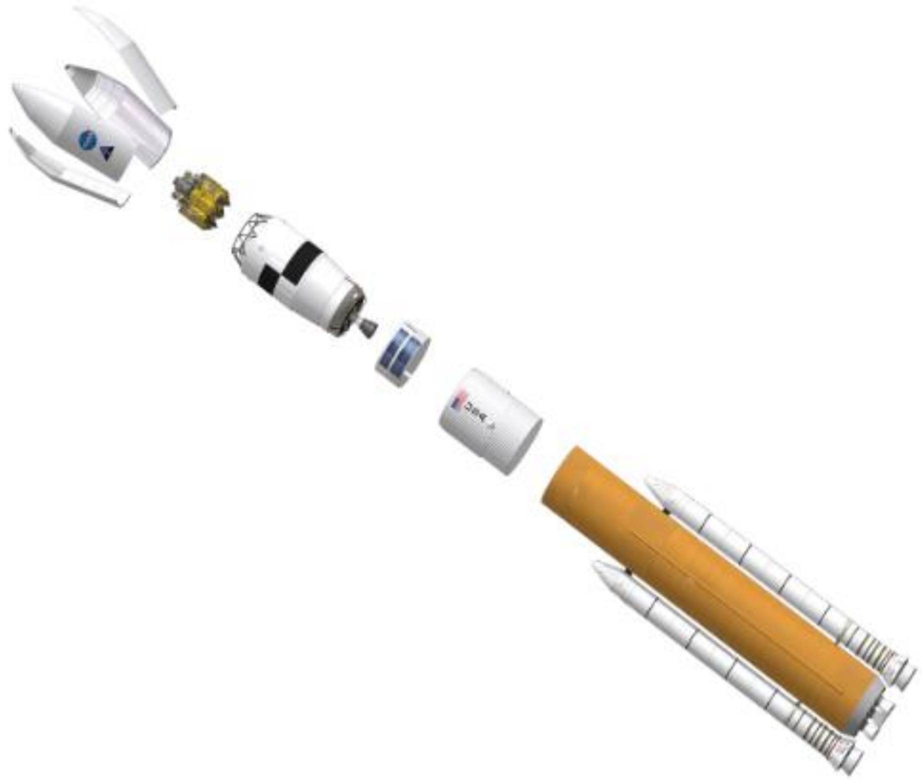
- **Integrated product(s)** with all system interactions identified and properly balanced.
- **Documentation and manuals** including system analysis models, data, and reports supporting flight-readiness rationale and available for future analysis during the operation of the system in the mission-execution phase.
- **Work products**, including reports, records, and non-deliverable outcomes of product integration activities (to support the Technical Data Management Process); integration strategy document; assembly/check area drawings; system/component documentation sequences and rationale for selected assemblies; interface management documentation; personnel requirements; special handling requirements; system documentation; shipping schedules; test equipment and drivers' requirements; emulator requirements; and identification of limitations for both hardware and software.

DSM – Design Structure Matrix

https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-842-fundamentals-of-systems-engineering-fall-2015/lecture-notes/MIT16_842F15_Ses_8_Sys_Int.pdf

IT IS IMPORTANT TO IDENTIFY THE INTERFACES.

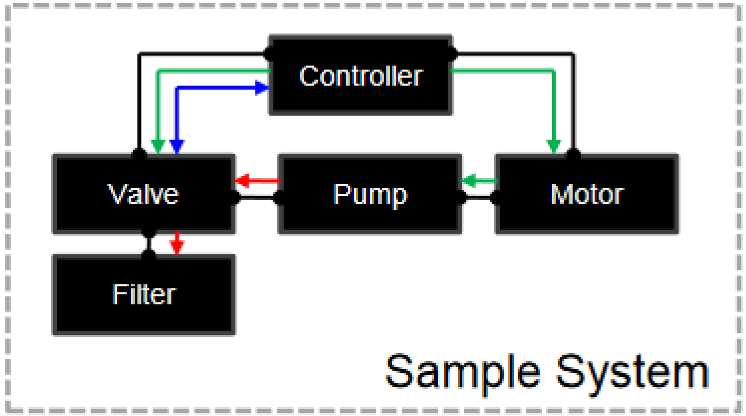
- Complex systems have many interfaces
  - Common interfaces reduce complexity
  - System architecture drives the types of interfaces to be utilized in the design process
  - Clear interface identification and definition reduces risk
  - Most of the problems in systems are at the interfaces.
  - Verification of all interfaces is critical for ensuring compatibility and operation



SYSTEMS ENGINEERING USUALLY DEALS WITH 4 TYPES OF INTERFACES

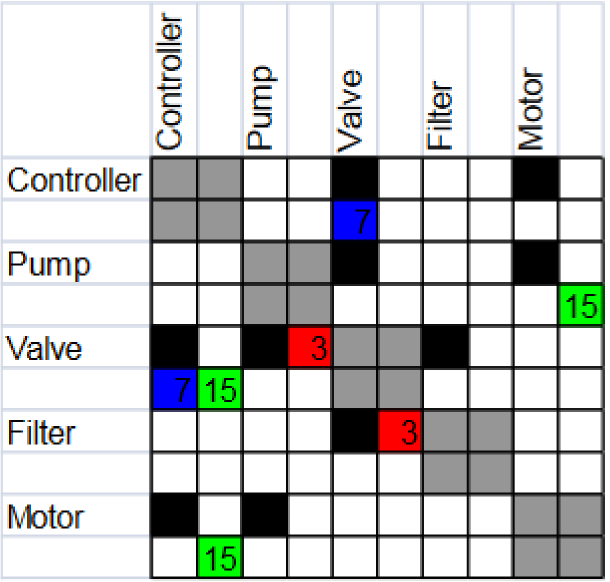
- Physical Connection
- Energy flow (electrical, termal, RF, mechanical)
- Mass flow (fluids, gases, solids)
- Information flow (data)

DSM CAPTURES CONNECTIVITY ARCHITECTURE



Sample System

Architecture Definition:
The embodiment of concept, and the allocation of physical/informational function (process) to elements of form (objects) and definition of structural interfaces among the objects



DSM

Number	Type	Flag
0	No Connection	0
1	Mechanical	
2	Flow	3
3	Information	7
4	Energy	15

Key

- DSM captures connectivity of components => architecture
- DSM provides analysis capability not present in a traditional schematic

1. Define which product or system to model
2. Assemble product documentation
3. Create product breakdown structure
4. Start a blank DSM spreadsheet for example in Excel
5. Label the rows and columns of the DSM with both an ID number 1 ...N and a component/subsystem name
6. Start by mapping all the physical connections in the system
7. Double-check the physical connections
8. Map out mass flows along physical connections
9. Double-check mass flows from start to origin
10. Map out the energy flows along physical connections
11. Double-check energy flows from start to finish
12. Map out information flows following physical connections
13. Double-check information flows in the system
14. Map interactions (flows) in the system that do not follow physical connections
15. Reorder the DSM to reveal “modules”
16. Double-check accuracy of DSM
17. Sign off and publish

Class Ending

Homework

- Create a DSM of our Satellite
  - It is one spreadsheet with the four interactions (physical, mass, energy, info)
  - Color by the subsystem

- Expectation: A Spreadsheet with all interactions among the five subsystems
  - Due: Oct 10th