# Systems Engineering Part 3

**Jesse Olson**

**University Nanosatellite Program Manager**

**AFRL/RV**

# Agenda

- What’s the point of this EAT?
- Assumptions
- Definitions
- Where are we?
- Unit-Level Specification
- Unit-Level Requirements vs. Specifications
- Capturing Unit-Level Specifications
- Performing Unit-Level Specifications
- Design Solution Definition
- UNP Avionics Development
- UNP Structural Development
- Want to learn more?

# References

Significant credit to Sam Baxendale, author of the NS-10 Systems Engineering EAT series from which this heavily drew. Also references:

- “UNP NS11 User’s Guide”, AFRL/RV, 2022
- “Applied Space Systems Engineering”, Larson/Kirkpatrick/Sellers/Thomas/Verma, 2009
- “Space Mission Engineering”, Wertz/Everett/Puschell, 2011
- “The NASA Systems Engineering Handbook”, NASA, 2007
- “INCOSE Systems Engineering Handbook”, INCOSE, 2010
- “Michigan Tech MEPIV Lecture: An Introduction to Systems Engineering”, King, 2019
- “UNP NS9 EAT: Systems Engineering”, Straight, 2016
- “ISO/IEC 15288 IEEE Systems Engineering Standard”, IEEE, 2015
- “ECSS-E-10A European Systems Engineering Standard”, ESA, 2018

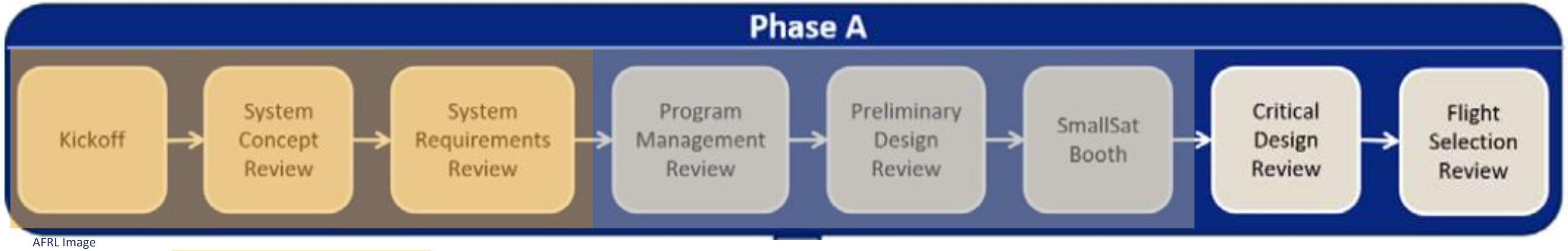
# What’s the point of this EAT?

**Today’s EAT will be slightly more ‘philosophical’ than previous SE EATs. But here are the main topics:**

1. How to transition from requirements definition to more detailed unit-level design?
2. Unit-level hardware selections must stay consistent with SE tools (CONOPS, Experiment Plan, RVM, System Budgets)
3. What are the expectations of avionics/structure development moving forward?

# Assumptions

- This EAT is relevant to activities *after* SRR
- Mission Design Document updated with feedback from SCR/SRR
- RVM defined from mission-level down to subsystem-level, feedback from SRR implemented
- Block diagrams at the mission- and system-level are currently in progress
- Design details of each subsystem, beyond subsystem requirements, remain unknown



Phase A

Kickoff → System Concept Review → System Requirements Review → Program Management Review → Preliminary Design Review → SmallSat Booth → Critical Design Review → Flight Selection Review

AFRL Image

**Complete-ish**

**Coming Soon**
(fast)

# Definitions

**Avionics:** All the electrical hardware and software that support the mission.

**Mission-level:** The space system(s) and ground system(s) cooperating as a “system of systems.”

**System-level:** Constituents of the mission – the space system (i.e. the spacecraft) or the ground system independently.

**Subsystem-level:** Constituents of a system – for example the space system has an Electrical Power Subsystem (EPS) or Attitude Determination & Control Subsystem (ADCS).

**Unit-level:** Constituents of a subsystem – a torque rod, battery module, processor PCB, interface PCB, reaction wheel, etc.

**Commercial-off-the-shelf (COTS):** Purchased from an industry vendor intended to be functional and ready to go “off-the-shelf.”

**In-house:** Designed and/or fabricated by the project team (e.g. a PCB designed by your team in Altium).

**Interface Control Document (ICD):** A mandatory UNP deliverable (Users Guide 8.3.4) that details hardware and software interfaces for each unit or subsystem.

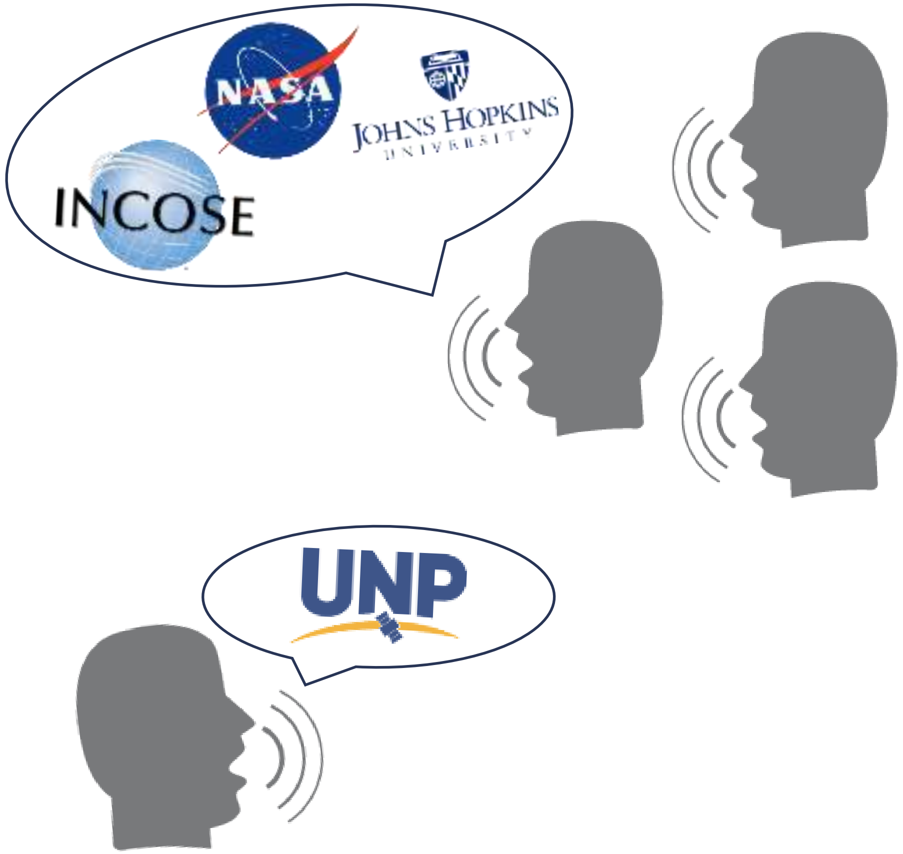
**Engineering-model (EM):** EM avionics are ideally an accurate representation of the flight hardware’s form, fit, and function. The EM is generally an independent set of hardware upon which most development occurs. Due to the stress induced on this hardware through extended testing and revisions, EM avionics are often considered unsuitable for flight.

**Flight-model (FM):** FM avionics are often thought of as the final revision of the EM and are intended for integration into the final satellite assembly.

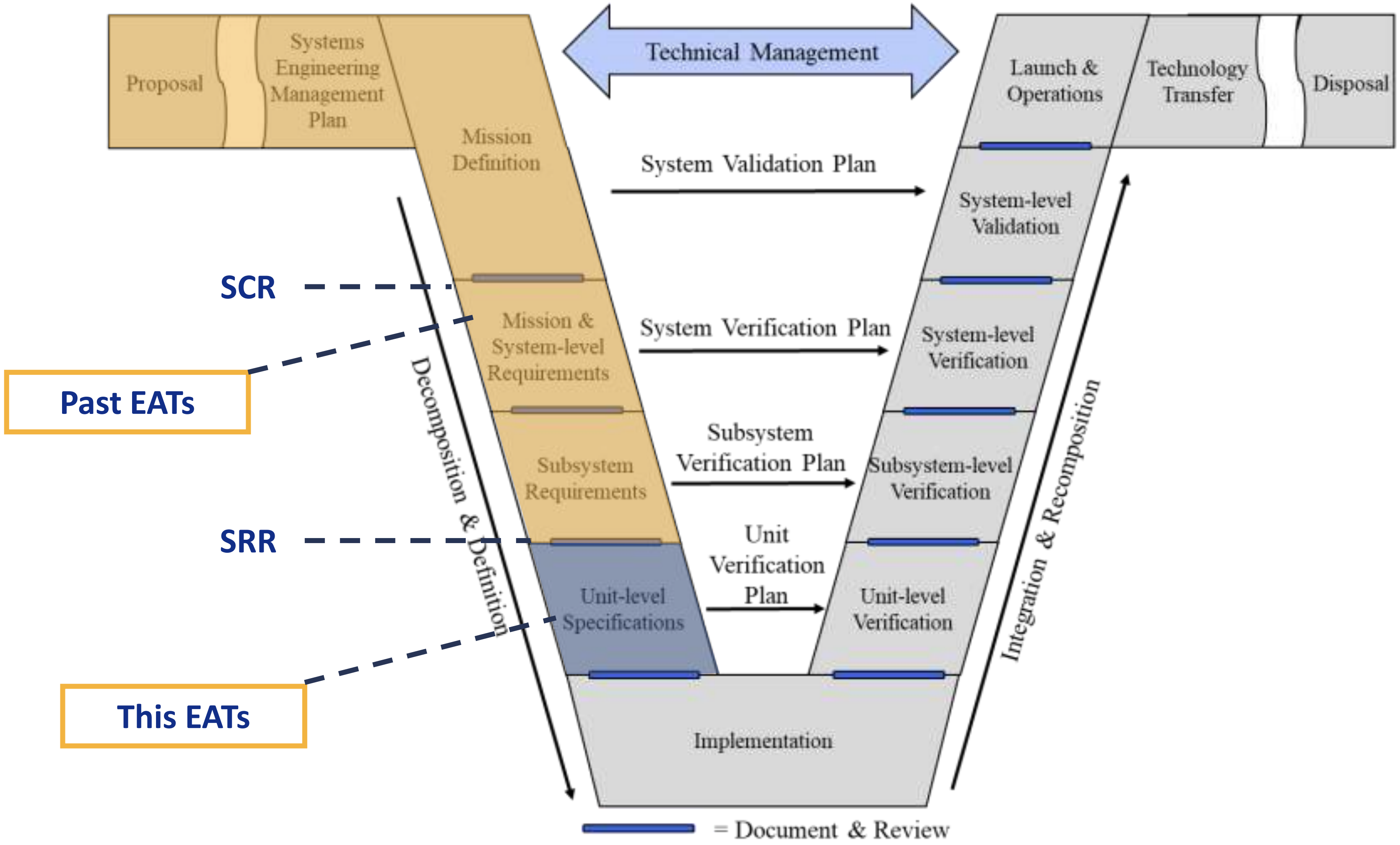
# UNP Disclaimer

**Systems Engineering comes in many flavors…**

- One organization’s best practices may differ from another’s
- Small satellites are a unique niche of the systems engineering community
  - There is a spectrum from large-scale production (e.g. Planet) to one-off R&D (e.g. AFRL)
- The point of UNP is to provide a *scoped* project framework to *introduce* students to Systems Engineering
  - Don’t get hung up on the details, often times just building and learning is the best approach
  - These EATs are intended to provide some context on why UNP is structured the way it is
- Is UNP the ideal Systems Engineering approach?
  - **No** (does an ideal SE even exist?). But it is a decent starting point.

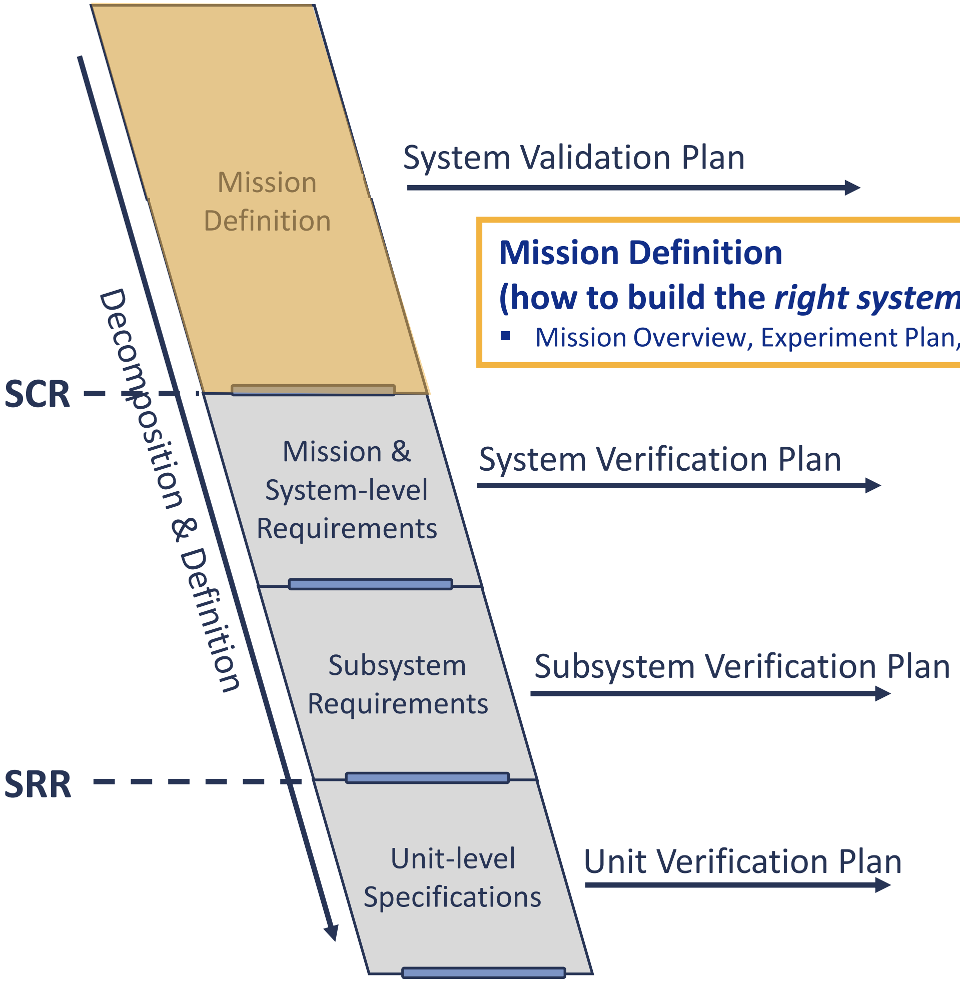


# Where are we?



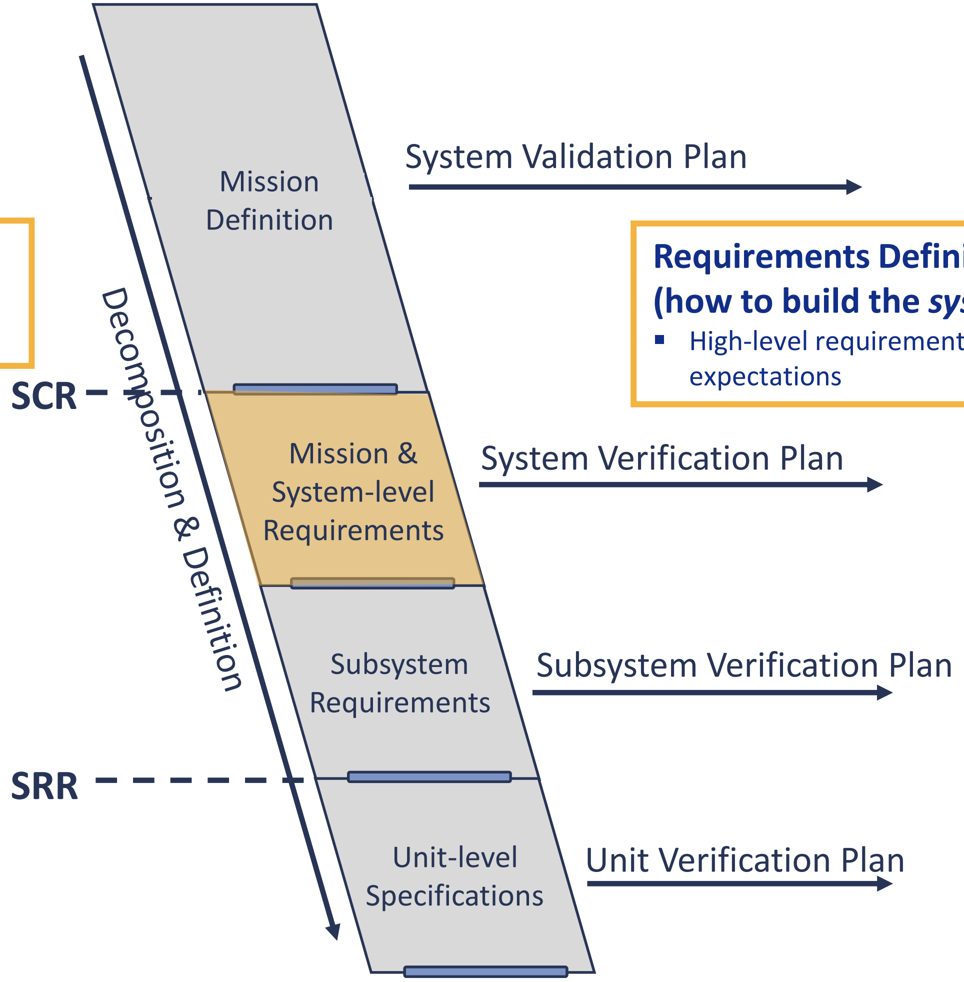
Approved for public release; distribution is unlimited. Public Affairs release approval AFRL-2023-1493

# Where are we?



**Mission Definition**
**(how to build the *right system*)**

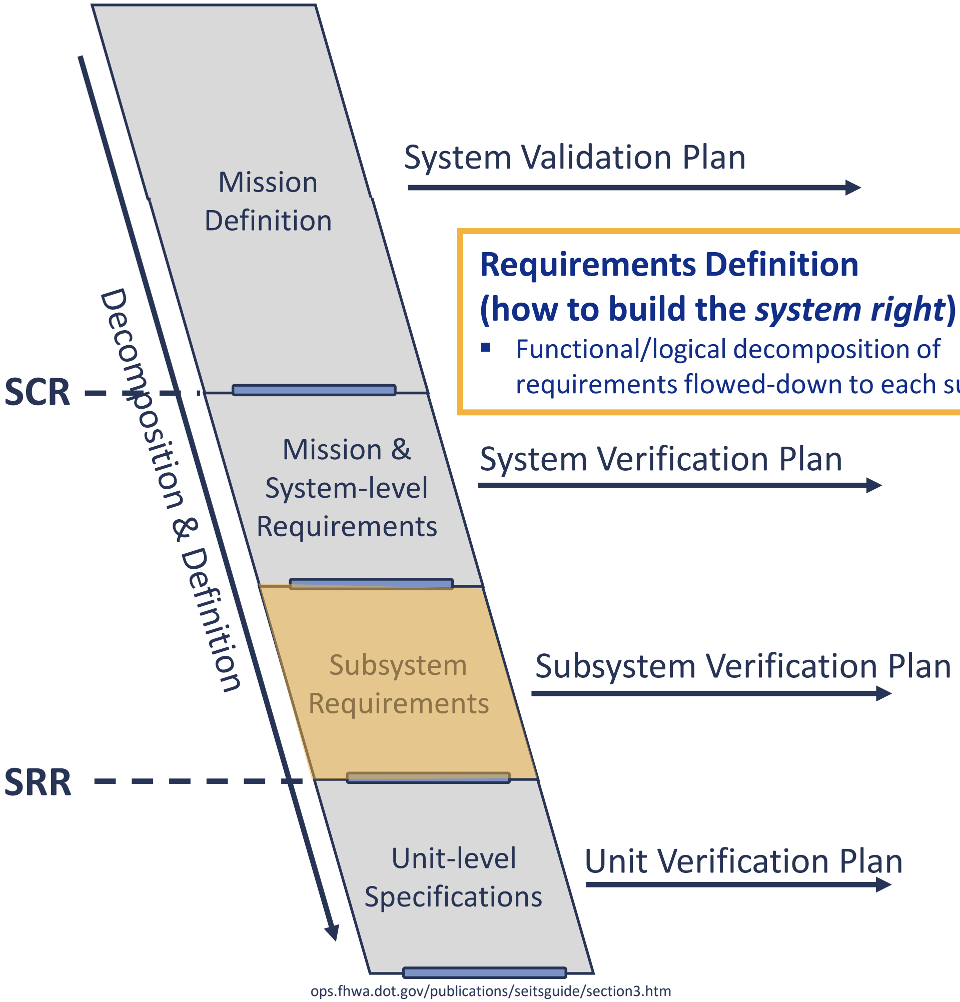
- Mission Overview, Experiment Plan, & CONOPS



**Requirements Definition**
**(how to build the *system right*)**

- High-level requirements derived from stakeholder expectations

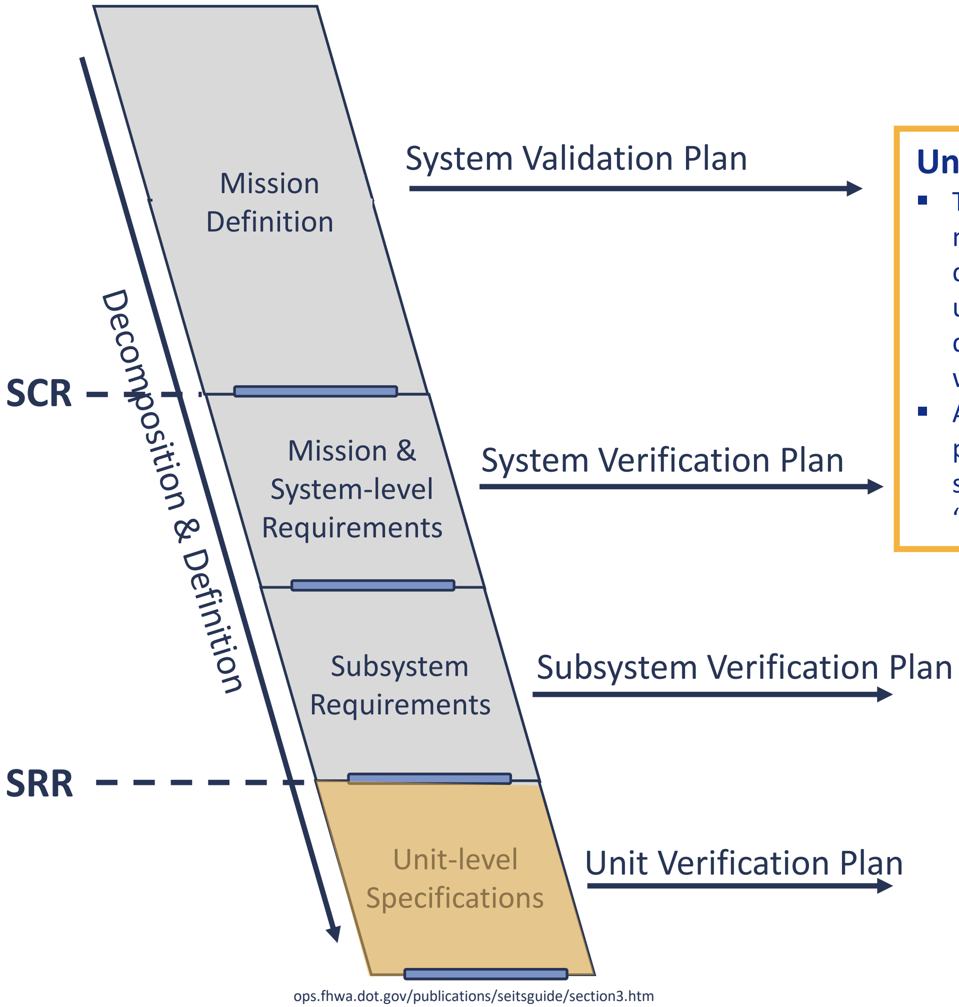
# Where are we?



ops.fhwa.dot.gov/publications/seitsguide/section3.htm

**Requirements Definition**
**(how to build the *system right*)**

- Functional/logical decomposition of requirements flowed-down to each subsystem

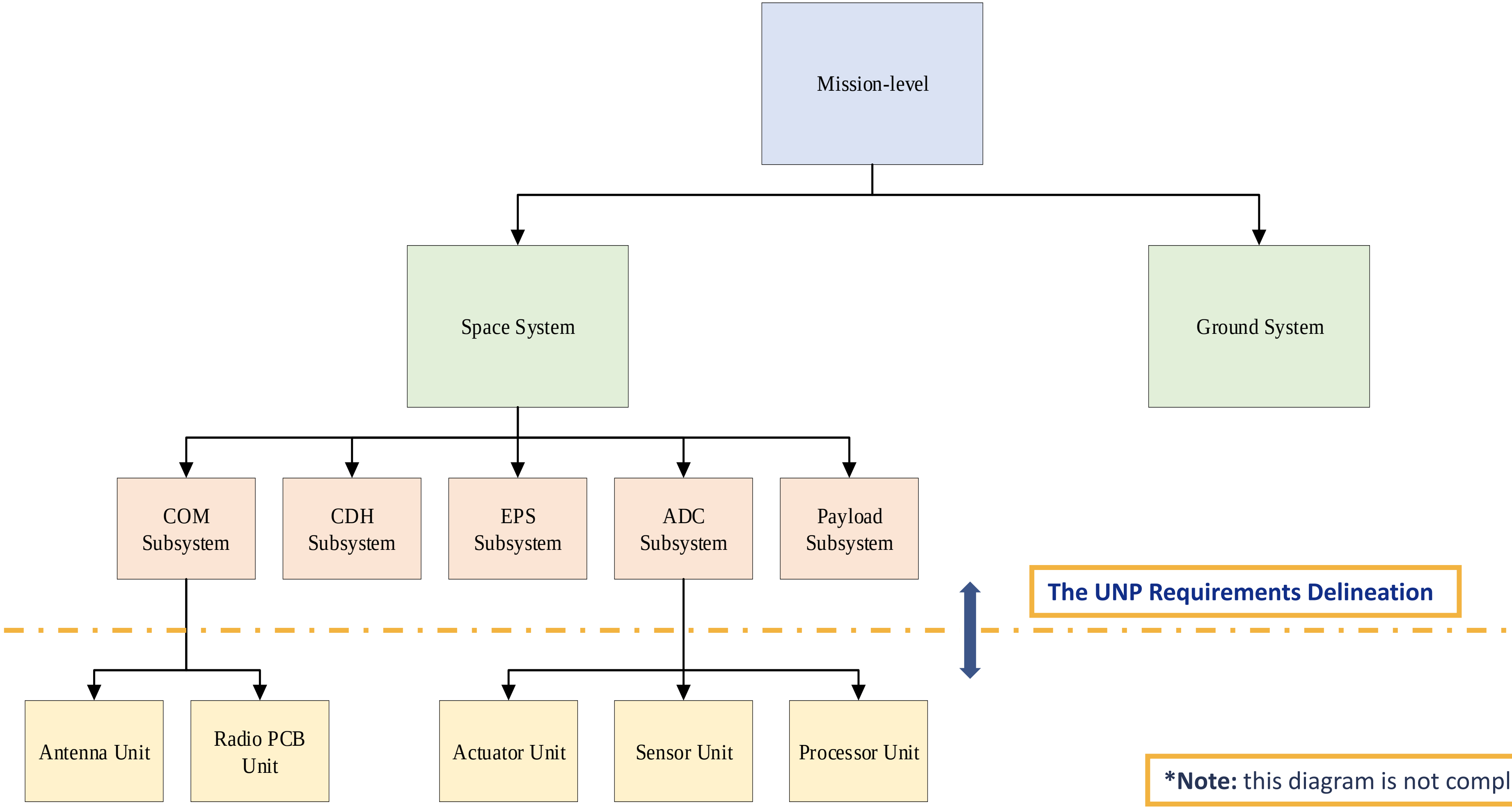


ops.fhwa.dot.gov/publications/seitsguide/section3.htm

**Unit-level Specification**

- Translate products from the mission & requirements definition processes into unit-level solutions that are designed to requirements within constraints
- Activities in this area of the project lifecycle are sometimes referred to as "Design Solution Definition"

# Unit-Level Specification



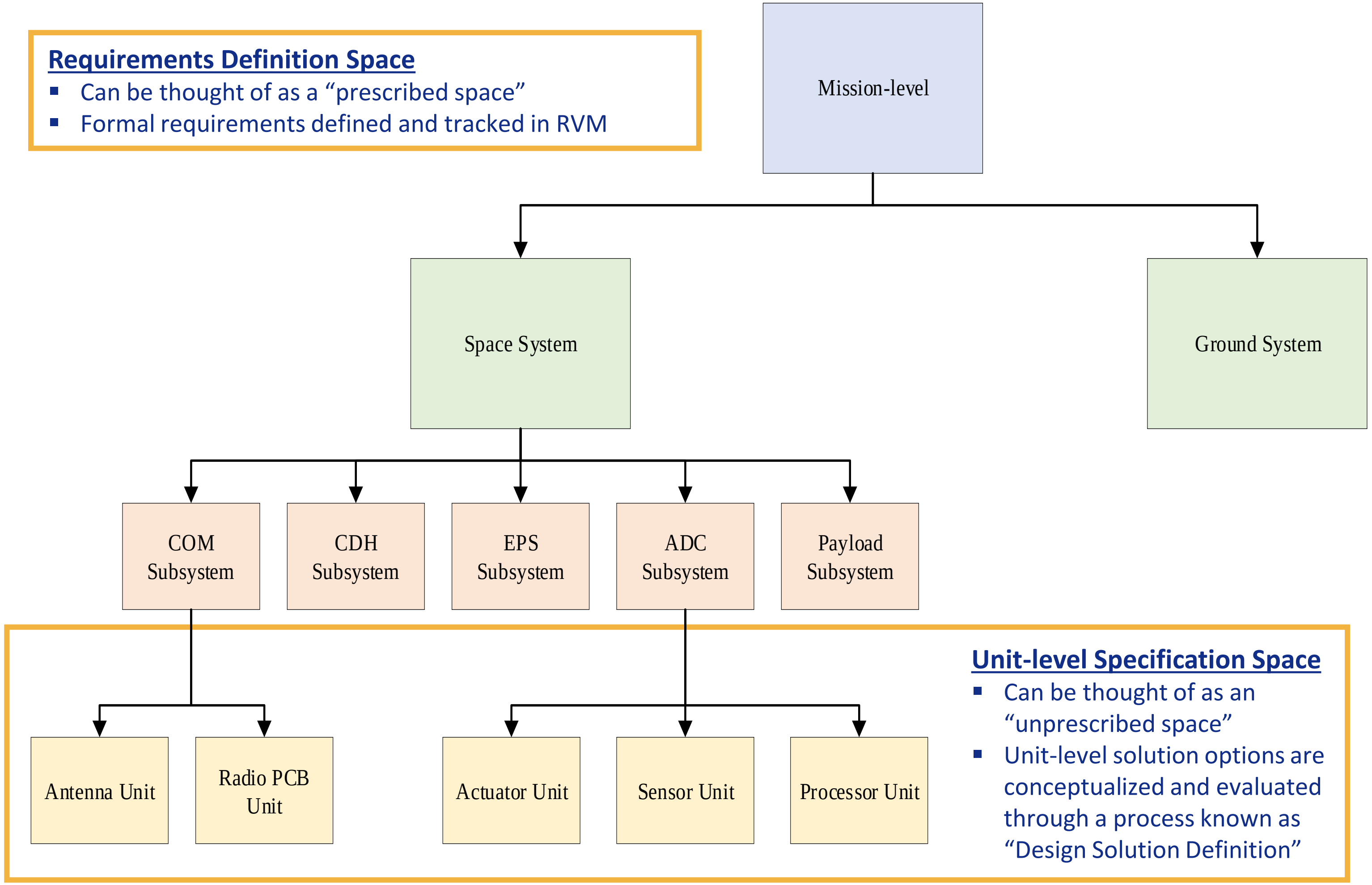
The UNP Requirements Delineation

***Note:** this diagram is not complete.

# Unit-Level Specification

**UNP teams typically only define requirements down to the subsystem-level in their RVM……**

- Below the subsystem-level, teams operate in a “Unit-level Specification” space in which they conduct trades to identify unit-level avionics capable of meeting subsystem-level requirements
- This specification must be conducted “within constraints” (what is realistic given UNP’s small budget, tight schedule, and limited technical expertise)
- Unit-level specification is where “the rubber meets the road”! The products of SCR and SRR are truly put through the wringer…



**Requirements Definition Space**

- Can be thought of as a “prescribed space”
- Formal requirements defined and tracked in RVM

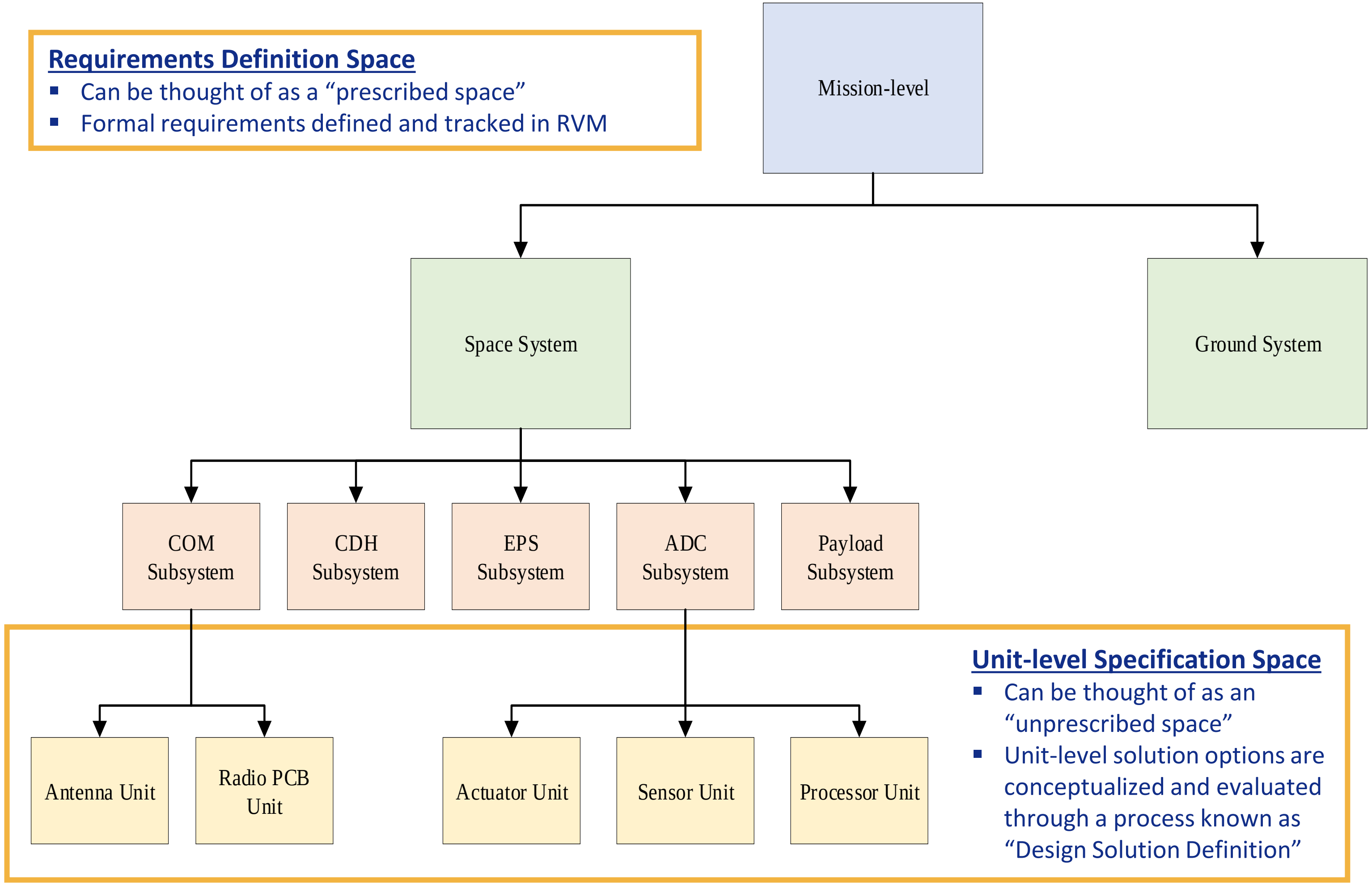
**Unit-level Specification Space**

- Can be thought of as an “unprescribed space”
- Unit-level solution options are conceptualized and evaluated through a process known as “Design Solution Definition”

# Unit-Level Specification

## A common pitfall in practical systems engineering is defining solutions “in a vacuum”

- Unit-level solution selections may impact the entire system, some may only impact the local subsystem, and **some may change your mission.**
- Simple example: the ADCS team evaluates and selects an actuator capable of meeting the mission’s strict attitude control requirement. However, the actuator is extremely power hungry.
  - Chief Engineer finds out months later when the Power Budget does not close (i.e. spacecraft is constantly power-negative).
- Effective engineers “push back” on bad requirements – if you are given an assignment to design/build/verify something that does not make sense. Speak up!



**Requirements Definition Space**
- Can be thought of as a “prescribed space”
- Formal requirements defined and tracked in RVM

**Unit-level Specification Space**
- Can be thought of as an “unprescribed space”
- Unit-level solution options are conceptualized and evaluated through a process known as “Design Solution Definition”

Approved for public release; distribution is unlimited. Public Affairs release approval AFRL-2023-1493

# Unit-Level Specification

**Design Solution Definition (i.e. UNP Unit-level Specification) can be an incredibly complex process……**

- Programs with thorough mission assurance strategies (e.g. legacy space system programs at NASA or Lockheed Martin) typically spend years evaluating thousands of design solutions - quantitively weighing these options against formal concept screening methodologies such as Pugh Matrices or Utility Theory
  - We don't have that luxury in UNP (nor is it really necessary)
- UNP teams only have a few months to consider a few options – **the key is to understand your constraints (e.g. cost, schedule, etc.) and make swift, informed decisions on what is the minimum effort necessary for full mission success**
- What does "informed decisions" mean? Maintain consistency between unit-level specifications and the Mission Design Document/RVM/System Budgets (e.g. Power, Link, Pointing, etc.)

# Unit-Level Requirements vs. Specification

## So, what is the difference between a requirement and a specification?

- Good question. This will be disputed until the end of time.
- Contemporary SE literature seems to agree on the following:
  - A requirement is a statement of what a product must do or a quality it must have
  - A specification is a collection of information that is imposed on the design and verification of a product; this includes all relevant requirements **and other information necessary for design, fabrication, and verification**
    - e.g. hardware interfaces, software interfaces, schematics/blueprints, dimensions, materials, etc.
  - A common analogy states that requirements are *inputs* to the design process while specifications are the *output*
- In industry, you will see all sorts of terms like ICDs, RVM, Requirements Specification, Design Specification, Qualification Report, Verification Report, Validation Report, etc. – often used interchangeably or with overlap from company to company
  - In UNP, you are required to submit Software Design Documents, Block Diagrams, ICDs, and an RVM (among other deliverables) – these are all pieces of what is referred to here as a specification
- **The point here is that, after SRR, UNP teams need to transition from requirements definition rather quickly. Specifications are your tool to translate requirements into design, build, integration, & test of avionics – eventually leading to a functioning FlatSat.**

# Capturing Unit-Level Specifications

## From UNP NS-11 User's Guide Chapter 2

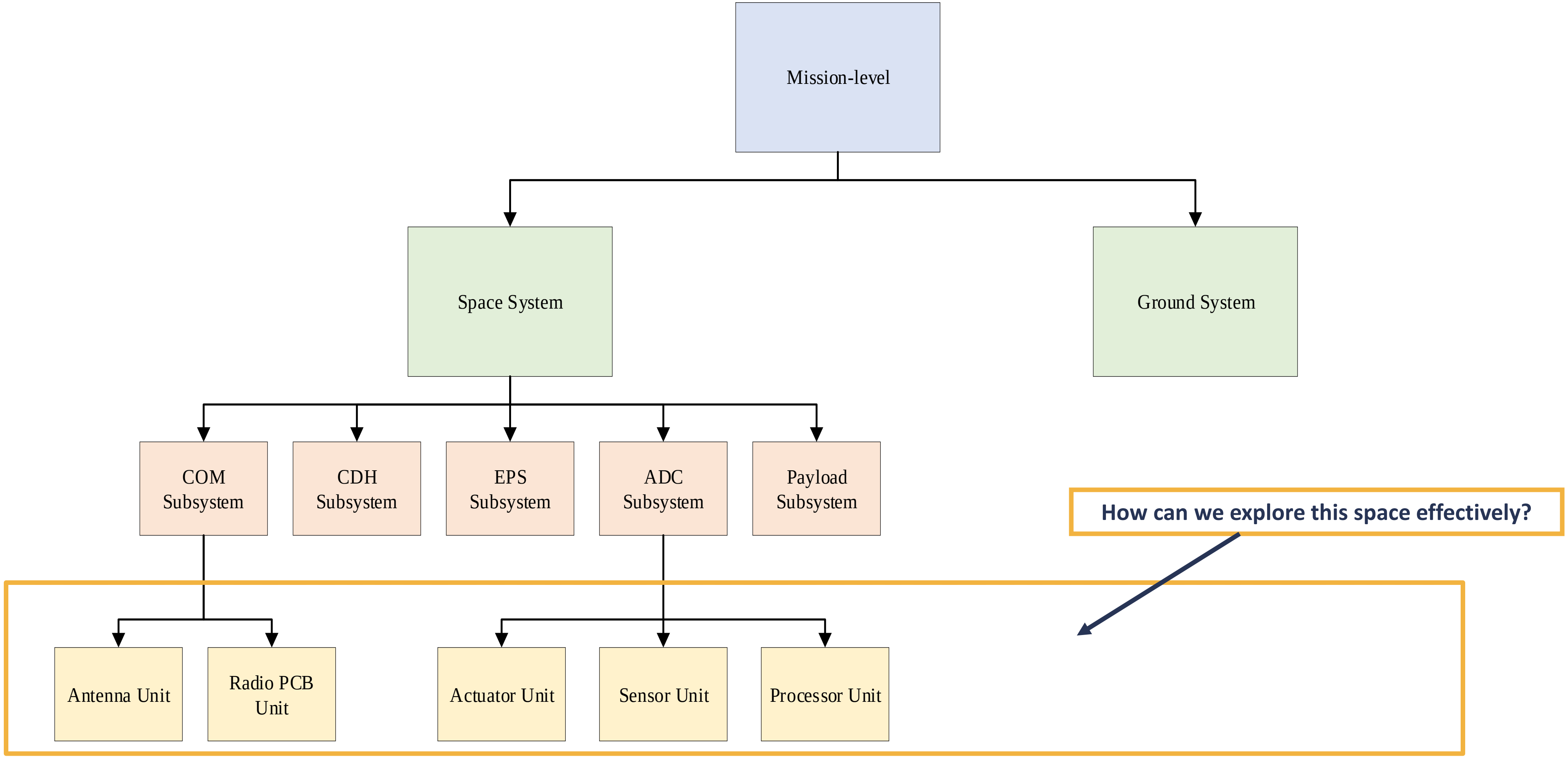
An effective Avionics Specification cannot be properly developed until the link between avionics development and requirements verification is first understood. The avionics development process is not just about designing hardware and writing code but verifying the design meet requirements. Recall that requirements are defined through a flow down from top (mission statement) to bottom (subsystem-level). The direction is reversed later in the satellite development process as requirements are verified from bottom to top. Avionics specifications bridge these major phases of the systems engineering process by imposing function and interface definitions at the unit-level to ensure requirements fulfillment at the subsystem and system-levels. This means the effectiveness of an avionics specification relies on the rigor of the requirements definition process. In other words: **teams may expend enormous resources (i.e. schedule, cost, personnel) developing the world's most advanced avionics bus, but it does not matter unless it meets mission-level requirements. It is only necessary to develop avionics sufficient enough to support the payload in meeting mission success.** Poor understanding of this relationship will ultimately result in excessive resource expenditure as teams struggle to integrate and verify their avionics.

# Capturing Unit-Level Specifications

## How should unit-level specifications be captured?

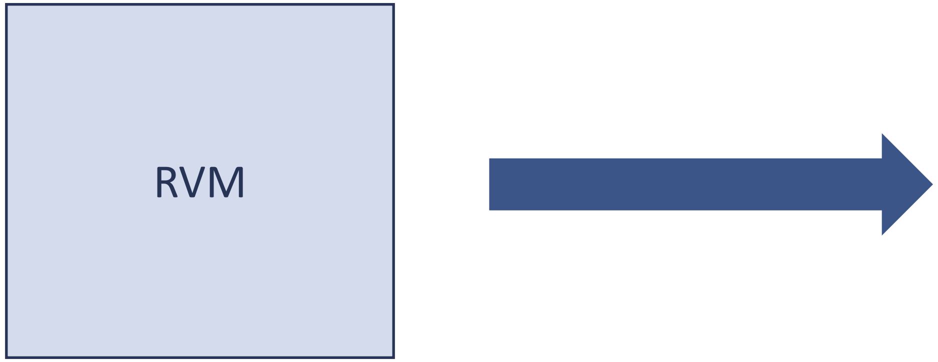
- Each UNP team will do it differently, for example some teams might:
  - Have a single “Subsystem Overview Document” that details the subsystem’s relevant requirements and block diagram. The document would also contain specifications for all the units that compose the subsystem
    - e.g. unit-level block diagrams, unit-to-unit hardware interfaces, dimensions, schematics, BOMs, software interfaces, etc.
  - Maintain a separate document for every avionics unit-level specification that composes their spacecraft
    - subsystem-level documents (e.g. subsystem-level block diagrams and requirements) are kept separate
  - Keep verification test procedures/results in separate documents than specifications
  - Millions of other ideas….
- Think about what works best for your team in terms of configuration management, revision tracking, and student turnover. All documentation should capture at least the following elements:
  - Design to requirements - How does the design of this unit tie to requirements verification at the subsystem, system, and mission-levels?
  - Functions - What functions must this unit perform to fulfill requirements?
  - Interfaces - How will this unit interface with other units/subsystems? Both hardware *and software* interfaces must be captured.
  - Note that this information is necessary regardless of the procurement strategy (COTS or in-house). If teams purchase a COTS unit that does not come with the appropriate data sheets and user manuals to answer these basic questions, there will be many headaches later on.

# Performing Unit-Level Specification



How can we explore this space effectively?

# Performing Unit-Level Specification



**ADC Subsystem**

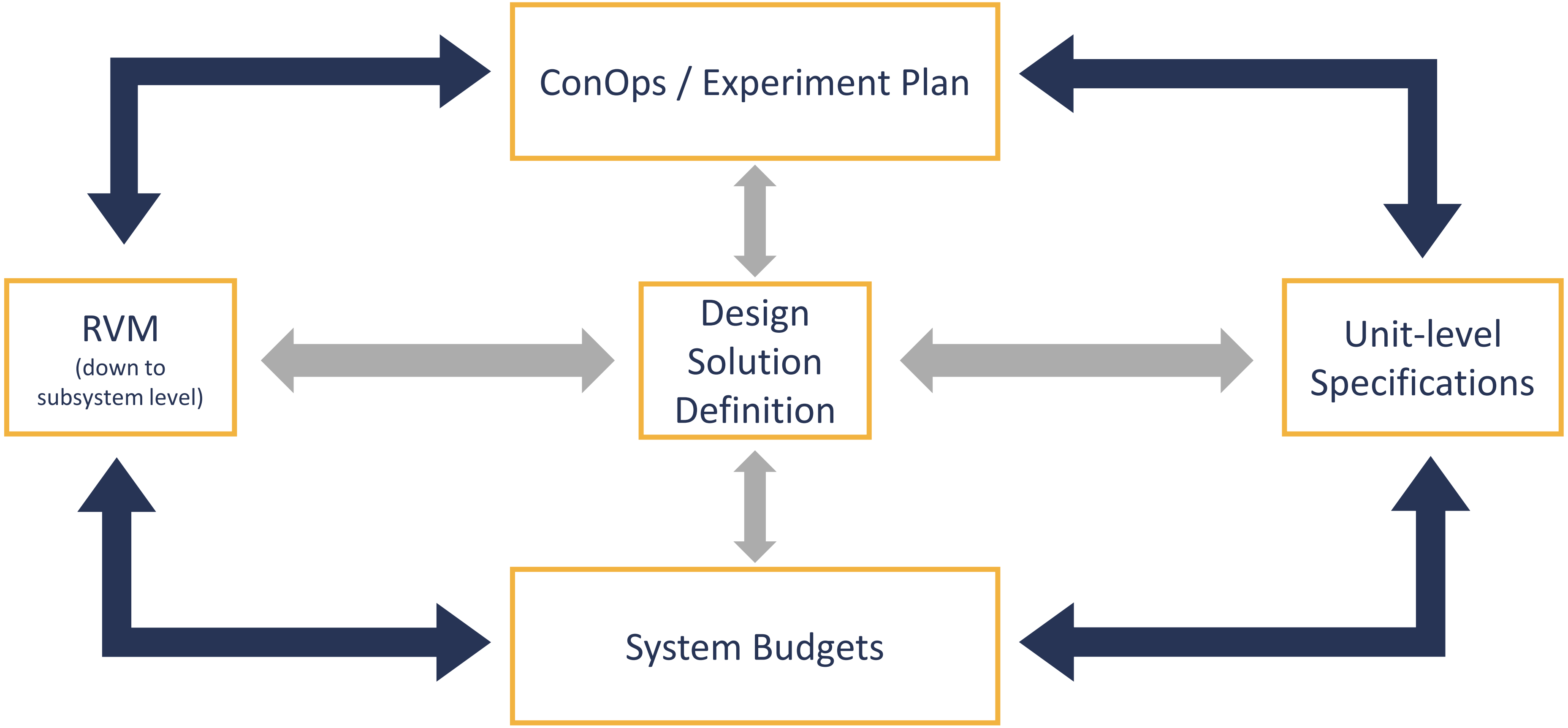
- Subsystem Block Diagram
- Torque Rod Unit Specification
- Reaction Wheel Unit Specification
- Gyroscope Unit Specification
- Magnetometer Unit Specification
- Star Tracker Unit Specification
- ADCS Processor PCB Unit Specification

**In other words, how do we get from here…..**

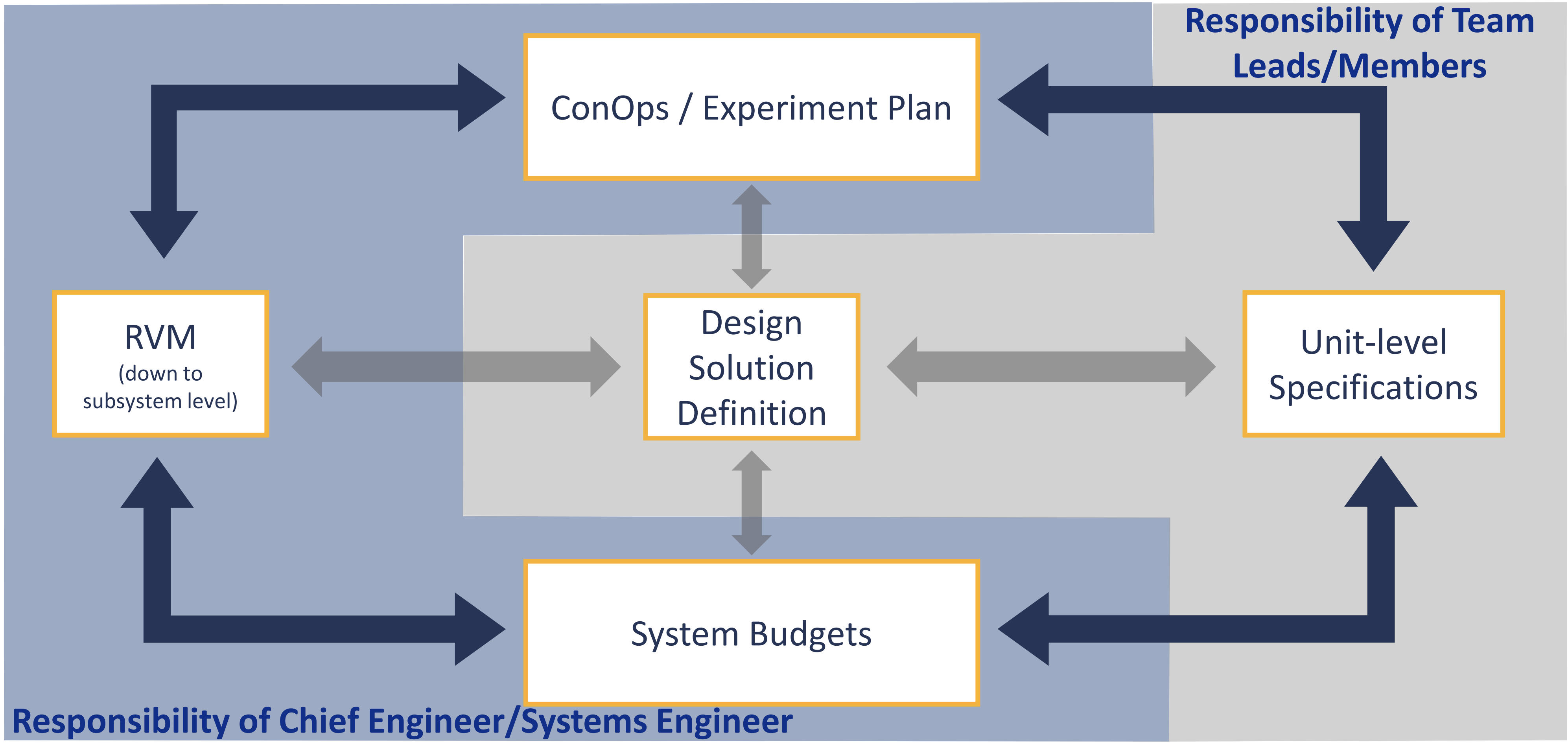
**…….to here?**

***Note:** used ADCS as an example here, but this could be applied to any system/subsystem/unit

# Performing Unit-Level Specification

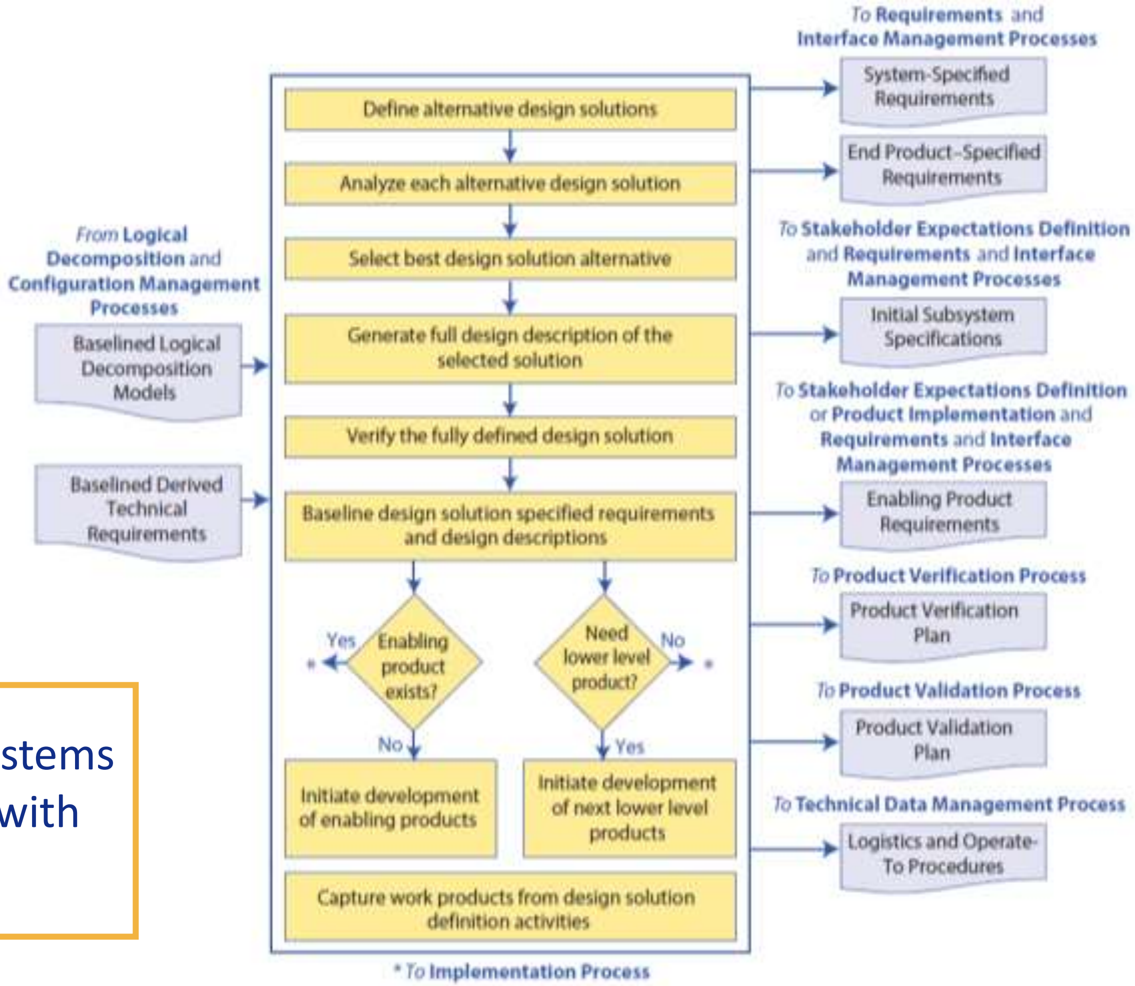


# Performing Unit-Level Specification



Responsibility of Team Leads/Members
ConOps / Experiment Plan
RVM
(down to subsystem level)
Design Solution Definition
Unit-level Specifications
System Budgets
Responsibility of Chief Engineer/Systems Engineer

# Design Solution Definition


To Requirements and Interface Management Processes
System-Specified Requirements
End Product-Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures
From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Enabling product exists?
Yes
*
No
Need lower level product?
No
*
Yes
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process

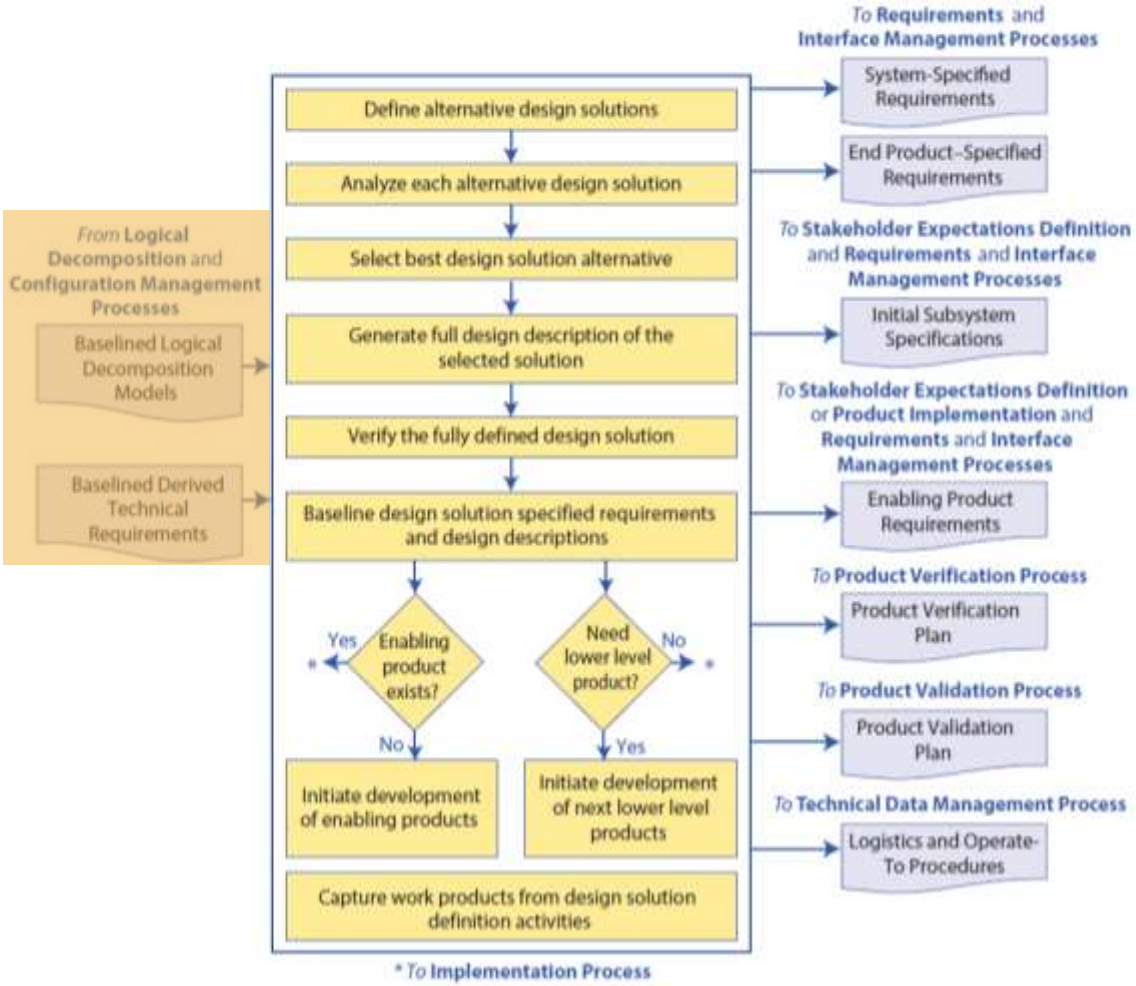
nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

According to the NASA Systems Engineering Handbook with some UNP context

***Note:** This process could be applied at any level of system hierarchy (system/subsystem/unit), we will focus on the unit-level application in this EAT (since you have already wrote subsystem-level requirements)

# Design Solution Definition

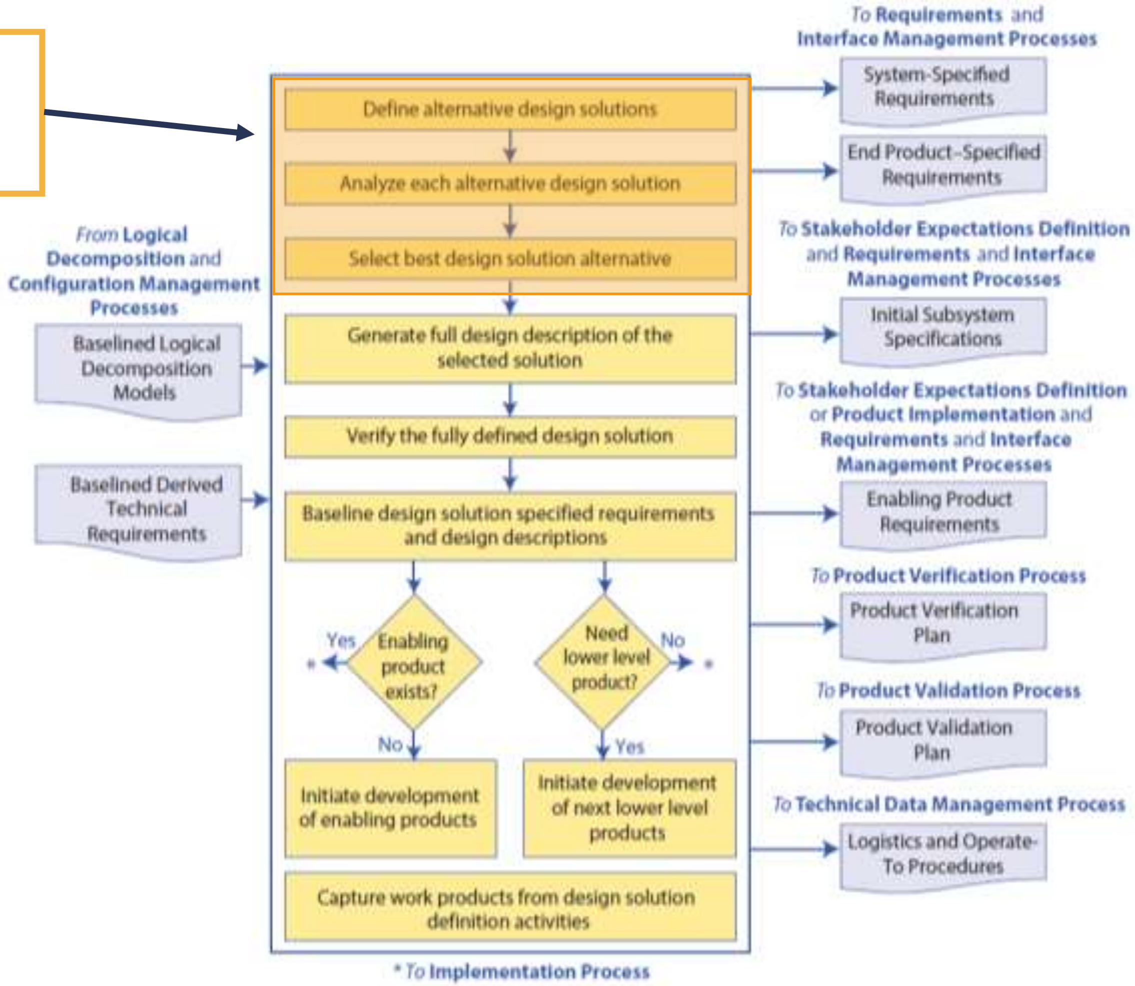
**Start:**
Mission Definition &
Requirements Definition


From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Yes
Enabling product exists?
No
*
Need lower level product?
No
Yes
*
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process
To Requirements and Interface Management Processes
System-Specified Requirements
End Product–Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

# Design Solution Definition

**Trade Studies**
- COTS vs. in-house
- Weighed against time, cost, & technical constraints



*From* Logical Decomposition and Configuration Management Processes

Baselined Logical Decomposition Models

Baselined Derived Technical Requirements

Define alternative design solutions

Analyze each alternative design solution

Select best design solution alternative

Generate full design description of the selected solution

Verify the fully defined design solution

Baseline design solution specified requirements and design descriptions

Enabling product exists?

Yes *

No

Initiate development of enabling products

Need lower level product?

No *

Yes

Initiate development of next lower level products

Capture work products from design solution definition activities

*To* Requirements and Interface Management Processes

System-Specified Requirements

End Product–Specified Requirements

*To* Stakeholder Expectations Definition and Requirements and Interface Management Processes

Initial Subsystem Specifications

*To* Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes

Enabling Product Requirements

*To* Product Verification Process

Product Verification Plan

*To* Product Validation Process

Product Validation Plan

*To* Technical Data Management Process

Logistics and Operate-To Procedures

* *To* Implementation Process

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

# Trade Studies

- Great to collect and document choices and evolution of mission
  - Can be at any level of design architectural (6U vs. 12U, Camera bs LIDAR), subsystem, vendor, part, etc.
- Generally, functionality/performance are traded against schedule, cost, or feasibility
  - Should outline MUST HAVE attributes vs. items that can be traded
  - Input should be as quantitative as possible; evaluation is often qualitative
- Often requires research, modeling, and programmatic evaluation to collect input to create trade

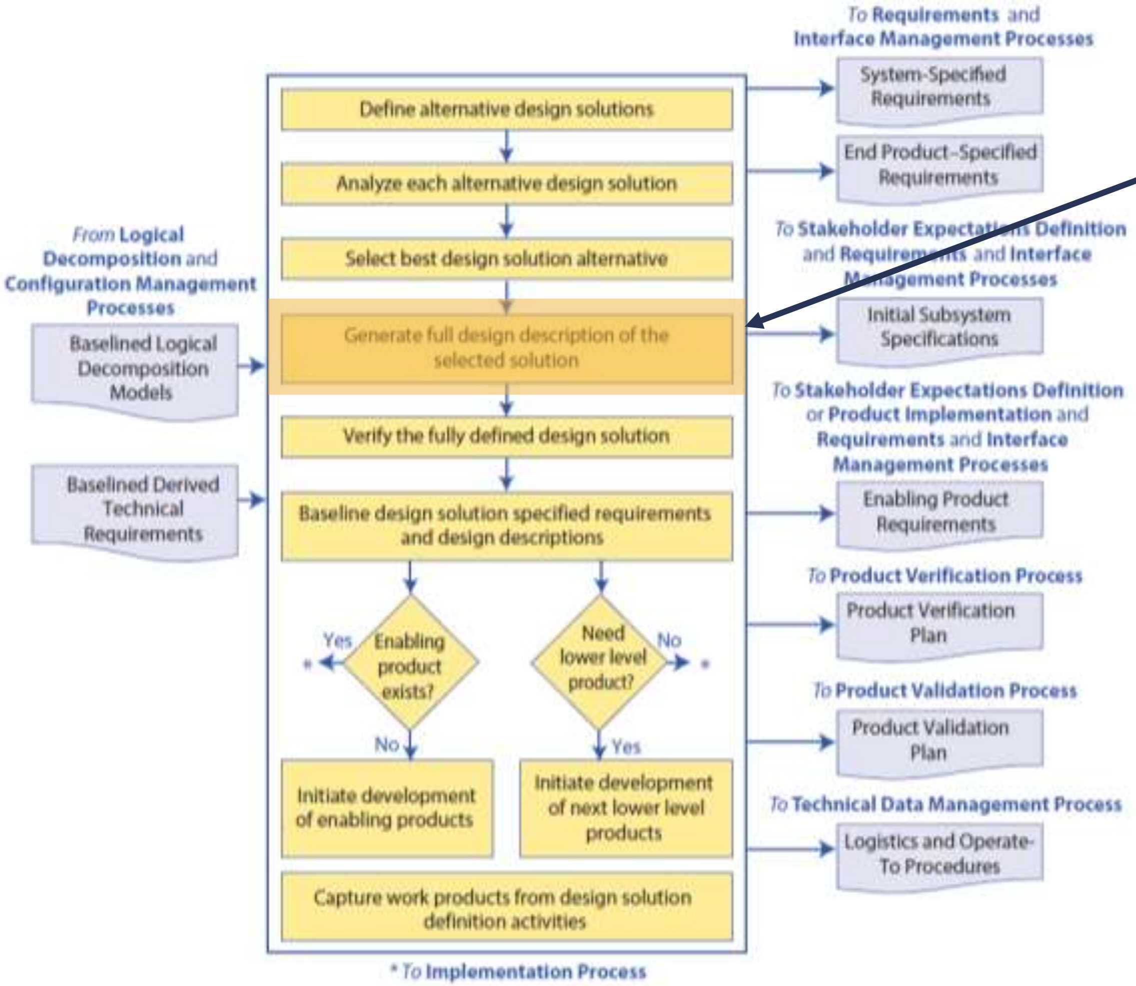
| Characteristic | Architecture/Part/Configuration #1 | Architecture/Part/Configuration #2 | Comments |
|---|---|---|---|
| Interface with other mission elements | Fits all interfaces | Requires some rework | ICD referenced |
| Performance | Achieves 70% of objectives | Achieves 90% of objectives | Based upon system budget |
| Lead Time (Schedule) | Integration with FlatSat in month 2 | Integration with FlatSat in month 2 | Both options meet schedule |
| Cost | $100 | $150 + $TBD for rework of interfaces | Both above desired cost but are possible. Rework is concern on #2 |

Approved for public release; distribution is unlimited. Public Affairs release approval AFRL-2023-1493

# Feasibility

- “Functional and Performance Analysis” + “Safe and Reliable, Affordable”
  - All of these are Feasibility studies
- Analyzing the Design Drivers is one of the keyways to determine feasibility
- System Budgets are great wat to assess (i.e. power, data, link, mass):
  - Iteration 1 = sizing/architecting of system + identifying drivers + feasibility
  - Iteration 2 = analysis tool to track progress/verification of these budgets with test results
  - Iteration 3 = Use as operations tool to ensure system is capable of a given operations profile
- Ideally, physics-based models of the technology or science demonstration exist to inform mission developers of key needs
- Utilize trade studies to compare capability vs. need vs. constraints (time, money, people knowledge)

# Design Solution Definition



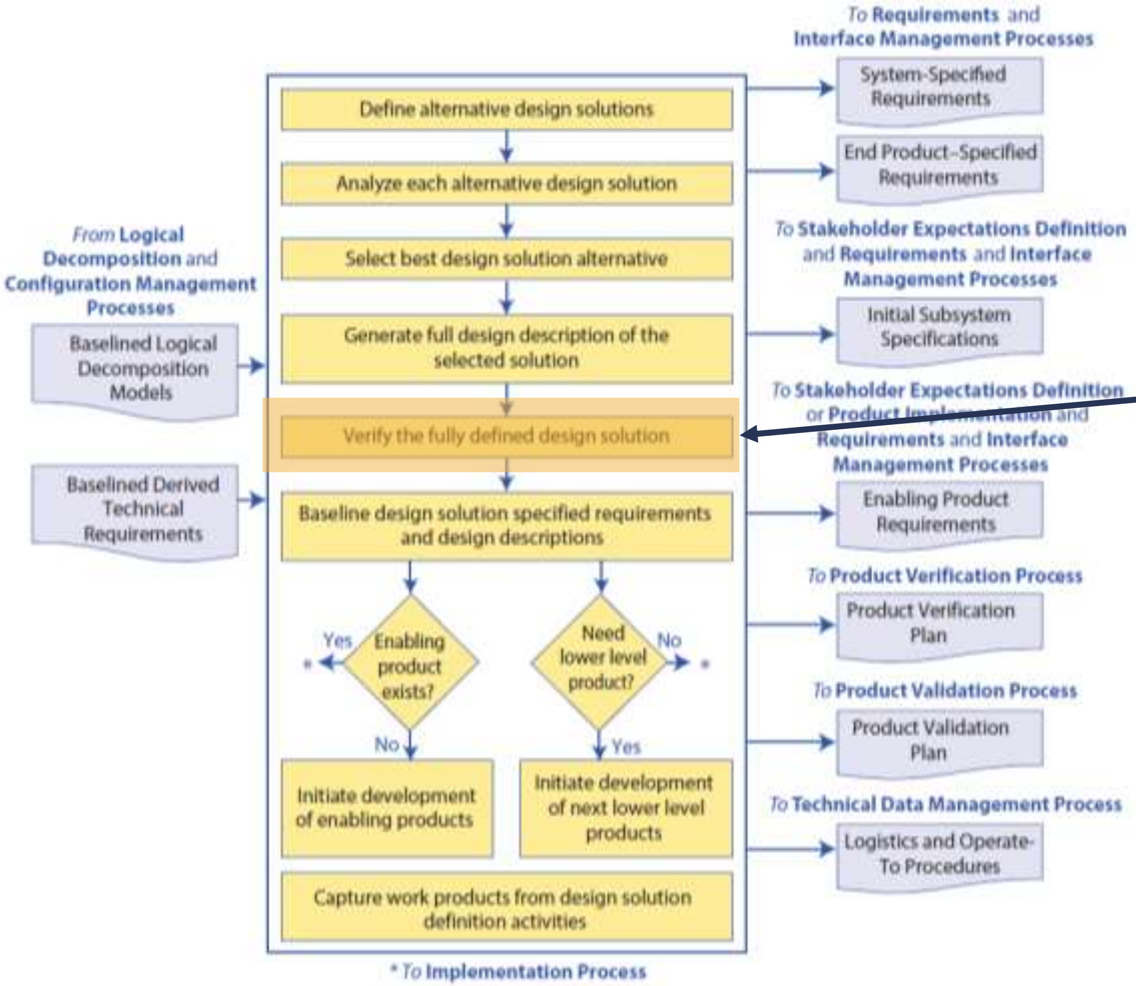
To Requirements and Interface Management Processes
System-Specified Requirements
End Product-Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures
From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Enabling product exists?
Yes
No
Need lower level product?
No
Yes
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

**Develop Unit-level Specifications**

- Block diagrams
- Functions
- ICDs
- Etc.

# Design Solution Definition



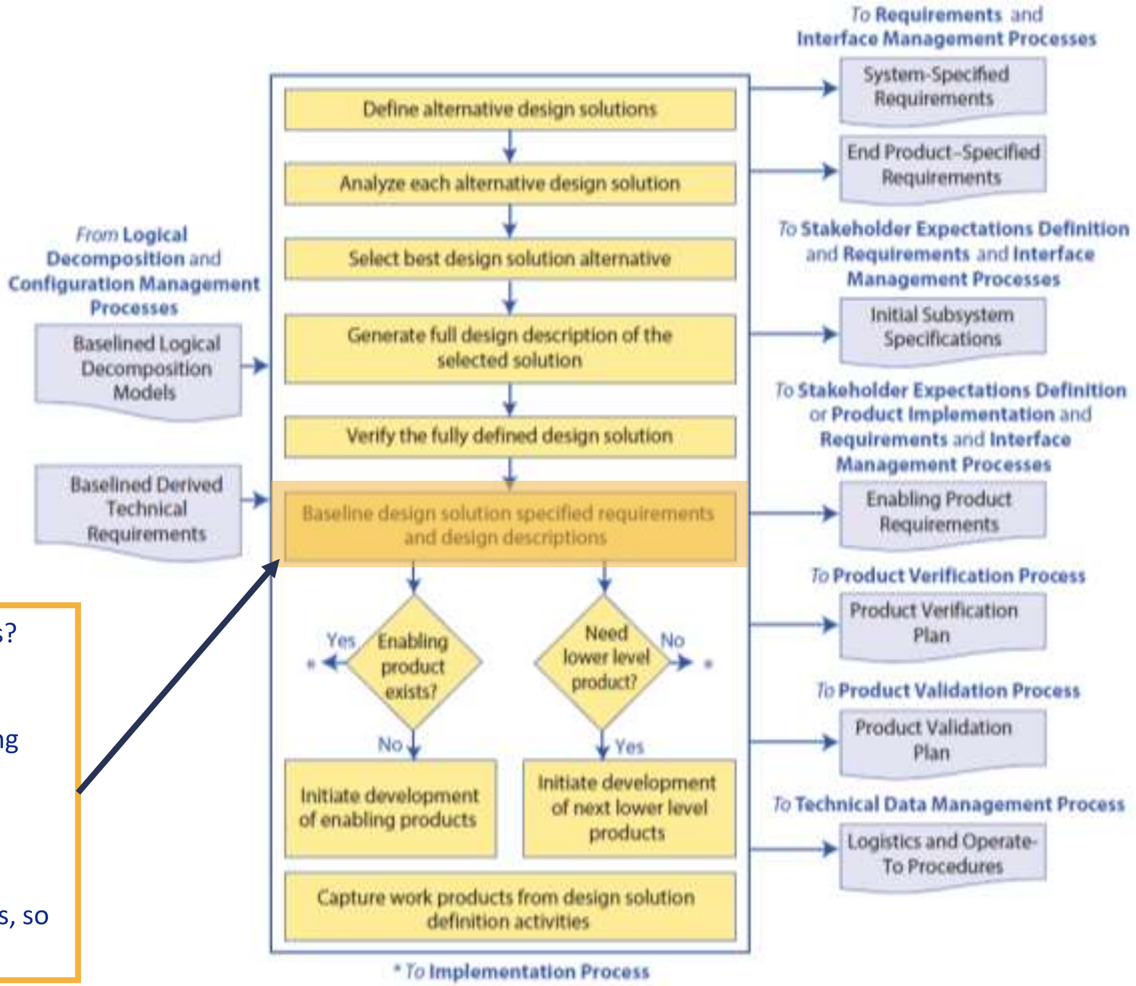
*To* Requirements and Interface Management Processes
System-Specified Requirements
End Product–Specified Requirements
*To* Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
*To* Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
*To* Product Verification Process
Product Verification Plan
*To* Product Validation Process
Product Validation Plan
*To* Technical Data Management Process
Logistics and Operate-To Procedures
*From* Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Enabling product exists?
Yes
No
Need lower level product?
No
Yes
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* *To* Implementation Process

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

You will do formal verification later through integration & test, but do some initial sanity checks

- Interface checks
- System budgets (power, data, etc.)
- Mod&sim (if applicable)
- Evaluation boards/initial testing

# Design Solution Definition


To Requirements and Interface Management Processes
System-Specified Requirements
End Product–Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures
From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Yes
Enabling product exists?
No
*
Need lower level product?
No
Yes
*
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

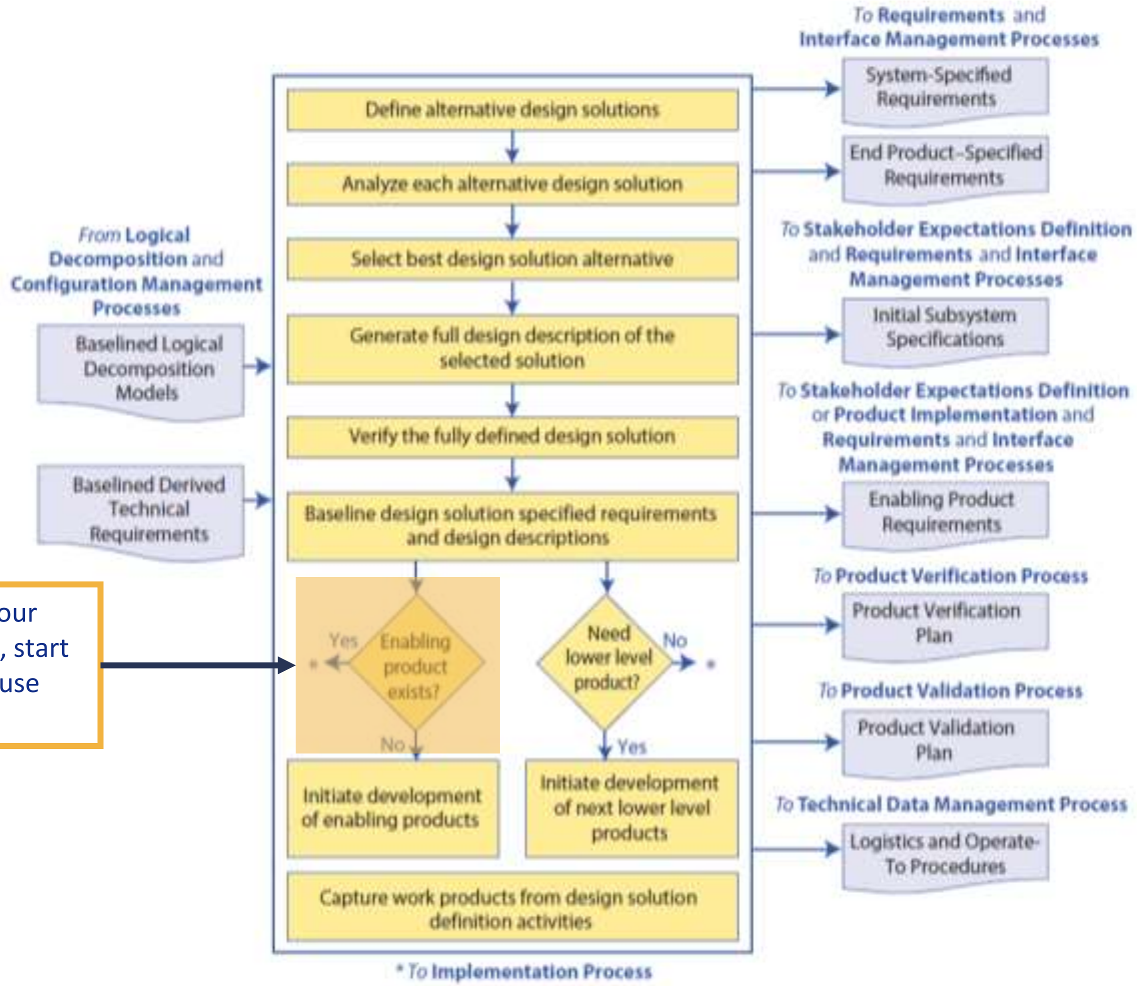
Will this design solution meet ALL requirements?

Again, you will formally verify this through integration & test later, but use your engineering judgement here based on the results from the previous step.

Remember UNP typically does not define requirements at the unit-level (NASA often does, so don’t get confused by this step)

# Design Solution Definition

Does a COTS solution exist that agrees with your cost, schedule, & technical constraints? If yes, start implementation. If no, start defining an in-house solution.



*From* **Logical Decomposition** and **Configuration Management Processes**

Baselined Logical Decomposition Models

Baselined Derived Technical Requirements

Define alternative design solutions

Analyze each alternative design solution

Select best design solution alternative

Generate full design description of the selected solution

Verify the fully defined design solution

Baseline design solution specified requirements and design descriptions

Enabling product exists?

Yes *

No

Initiate development of enabling products

Need lower level product?

No *

Yes

Initiate development of next lower level products

Capture work products from design solution definition activities

* *To* **Implementation Process**

*To* **Requirements** and **Interface Management Processes**

System-Specified Requirements

End Product–Specified Requirements

*To* **Stakeholder Expectations Definition** and **Requirements** and **Interface Management Processes**

Initial Subsystem Specifications

*To* **Stakeholder Expectations Definition** or **Product Implementation** and **Requirements** and **Interface Management Processes**

Enabling Product Requirements

*To* **Product Verification Process**

Product Verification Plan

*To* **Product Validation Process**

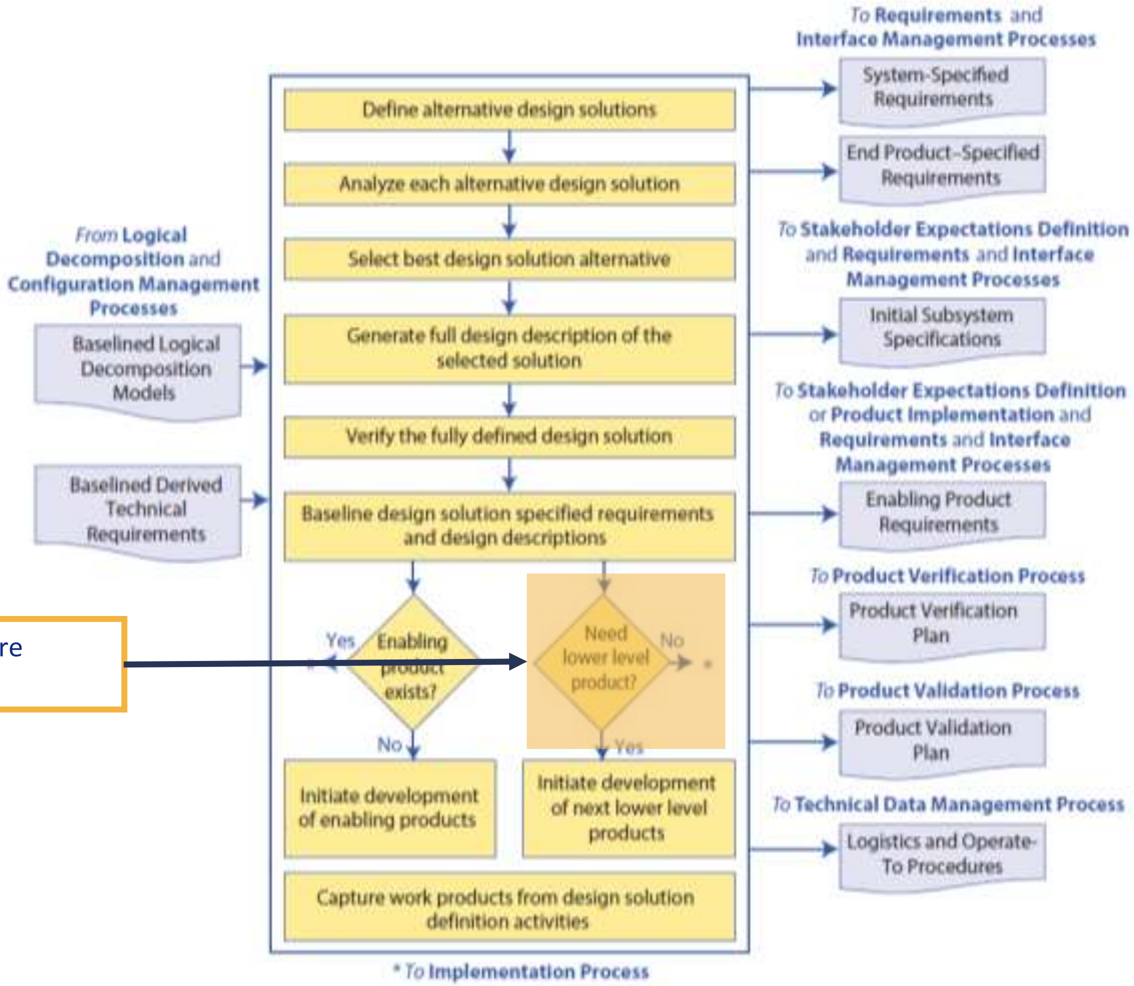
Product Validation Plan

*To* **Technical Data Management Process**

Logistics and Operate-To Procedures

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

# Design Solution Definition



*From* **Logical Decomposition** and **Configuration Management Processes**

Baselined Logical Decomposition Models

Baselined Derived Technical Requirements

Define alternative design solutions

Analyze each alternative design solution

Select best design solution alternative

Generate full design description of the selected solution

Verify the fully defined design solution

Baseline design solution specified requirements and design descriptions

Enabling product exists?

Yes

No

Need lower level product?

No

Yes

Initiate development of enabling products

Initiate development of next lower level products

Capture work products from design solution definition activities

* *To* **Implementation Process**

*To* **Requirements** and **Interface Management Processes**

System-Specified Requirements

End Product–Specified Requirements

*To* **Stakeholder Expectations Definition** and **Requirements** and **Interface Management Processes**

Initial Subsystem Specifications

*To* **Stakeholder Expectations Definition** or **Product Implementation** and **Requirements** and **Interface Management Processes**

Enabling Product Requirements

*To* **Product Verification Process**

Product Verification Plan

*To* **Product Validation Process**

Product Validation Plan

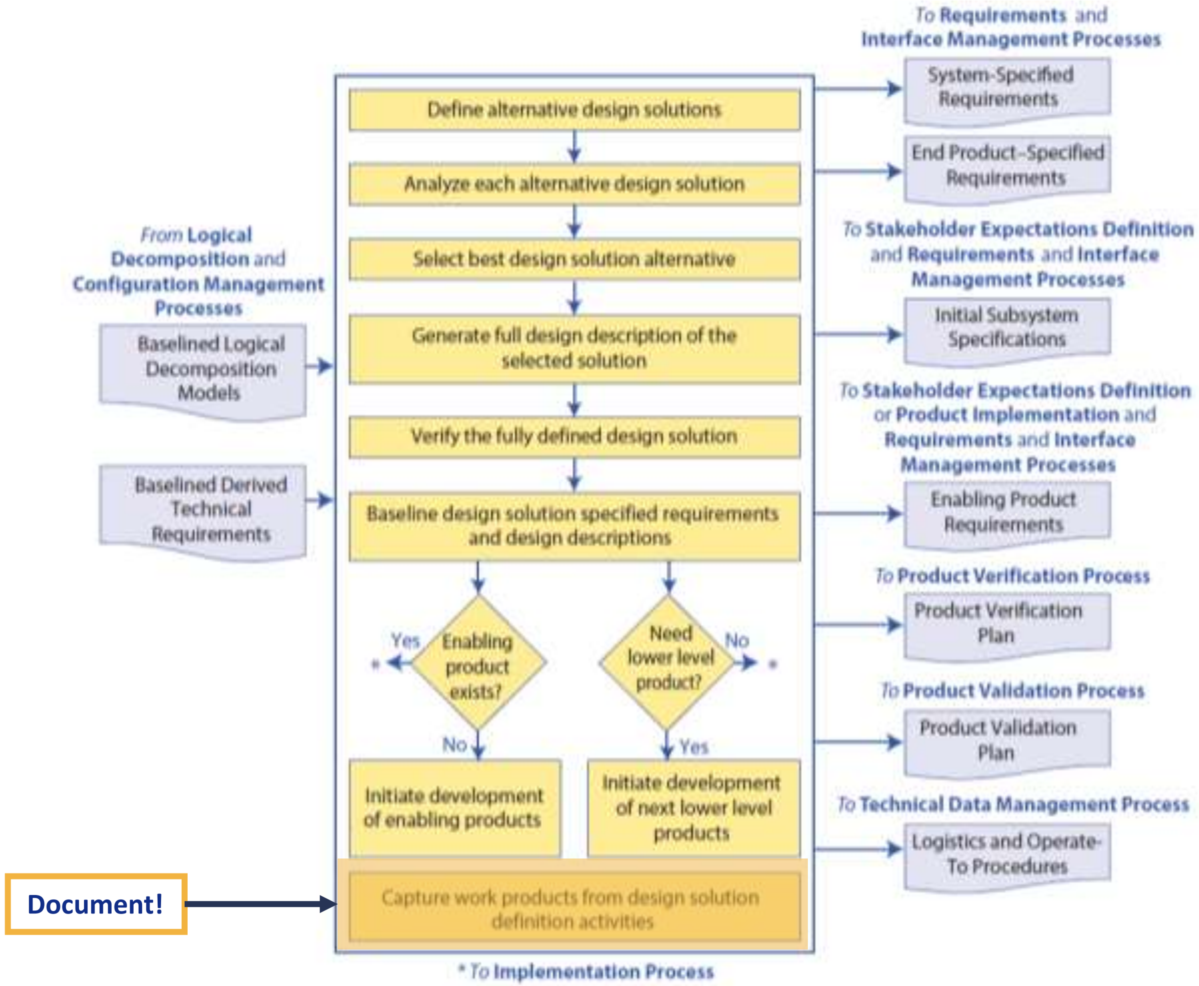
*To* **Technical Data Management Process**

Logistics and Operate-To Procedures

Is further decomposition required? Are there missing requirements?

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

# Design Solution Definition

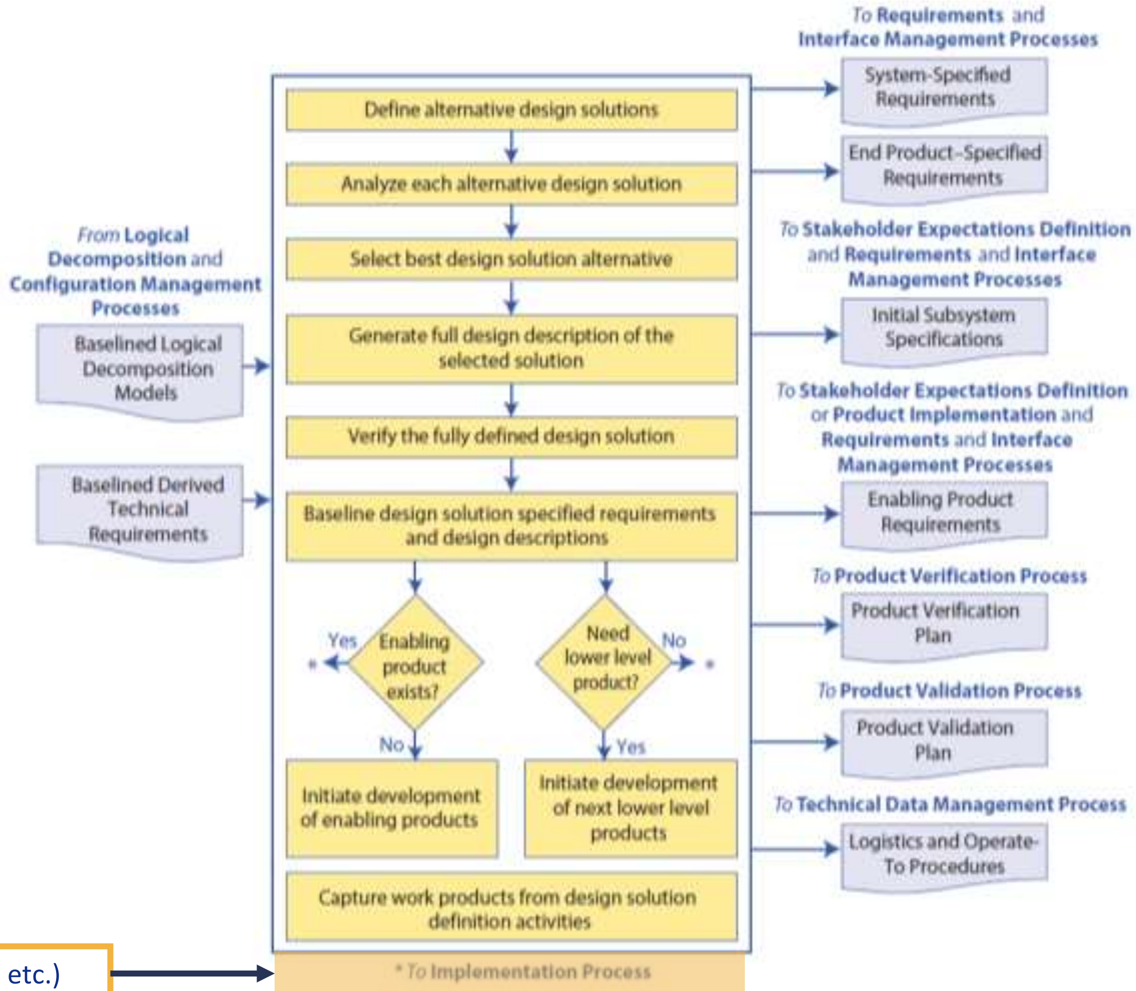


To Requirements and Interface Management Processes
System-Specified Requirements
End Product–Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures
From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Enabling product exists?
Yes
No
*
Need lower level product?
No
Yes
*
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process

**Document!**

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

# Design Solution Definition

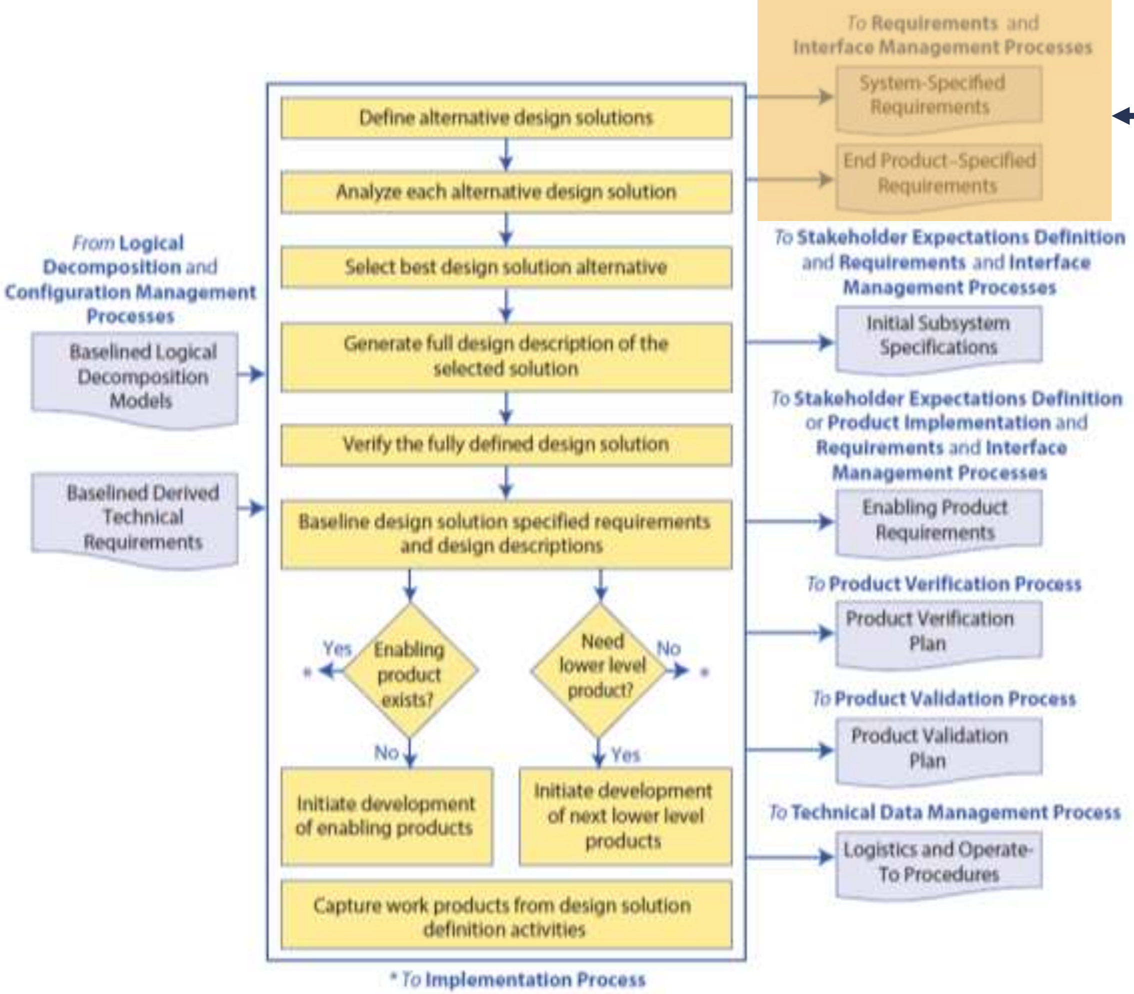

From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Enabling product exists?
Yes
No
*
Need lower level product?
No
Yes
*
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process
To Requirements and Interface Management Processes
System-Specified Requirements
End Product–Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures

Bottom of the "V" - (CAD, PCBs, Code, etc.)

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

Approved for public release; distribution is unlimited. Public Affairs release approval AFRL-2023-1493

# Design Solution Definition



*To* Requirements and Interface Management Processes

System-Specified Requirements

End Product–Specified Requirements

*From* Logical Decomposition and Configuration Management Processes

Baselined Logical Decomposition Models

Baselined Derived Technical Requirements

Define alternative design solutions

Analyze each alternative design solution

Select best design solution alternative

Generate full design description of the selected solution

Verify the fully defined design solution

Baseline design solution specified requirements and design descriptions

Enabling product exists?

Yes *

No

Initiate development of enabling products

Need lower level product?

No *

Yes

Initiate development of next lower level products

Capture work products from design solution definition activities

* *To* Implementation Process

*To* Stakeholder Expectations Definition and Requirements and Interface Management Processes

Initial Subsystem Specifications

*To* Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes

Enabling Product Requirements

*To* Product Verification Process

Product Verification Plan

*To* Product Validation Process

Product Validation Plan

*To* Technical Data Management Process

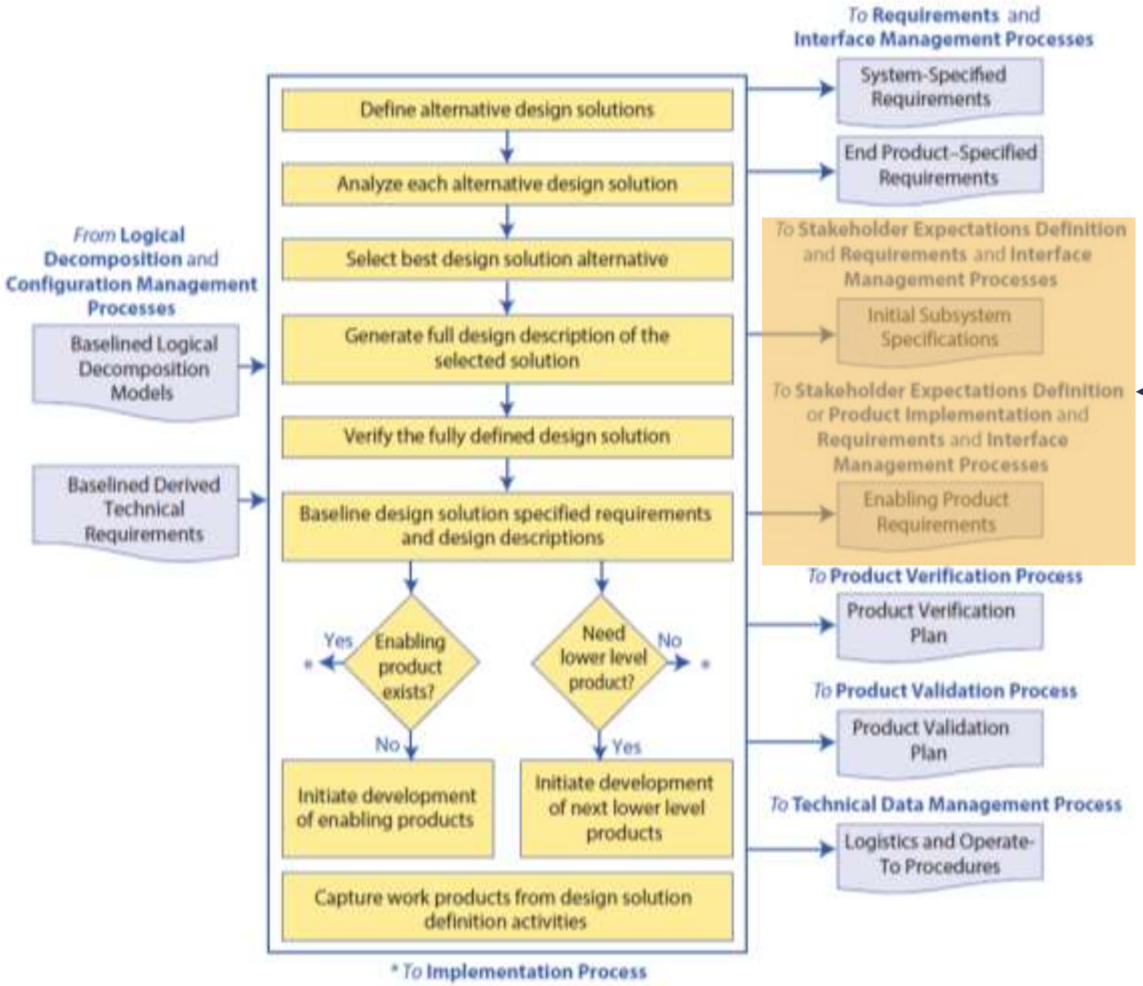
Logistics and Operate-To Procedures

nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

**Create a feedback loop between design solution options and your RVM**

- Are design solutions scoped within requirements?
- Did you miss any requirements?

# Design Solution Definition



To Requirements and Interface Management Processes
System-Specified Requirements
End Product–Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures
From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Enabling product exists?
Yes
No
*
Need lower level product?
No
Yes
*
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process

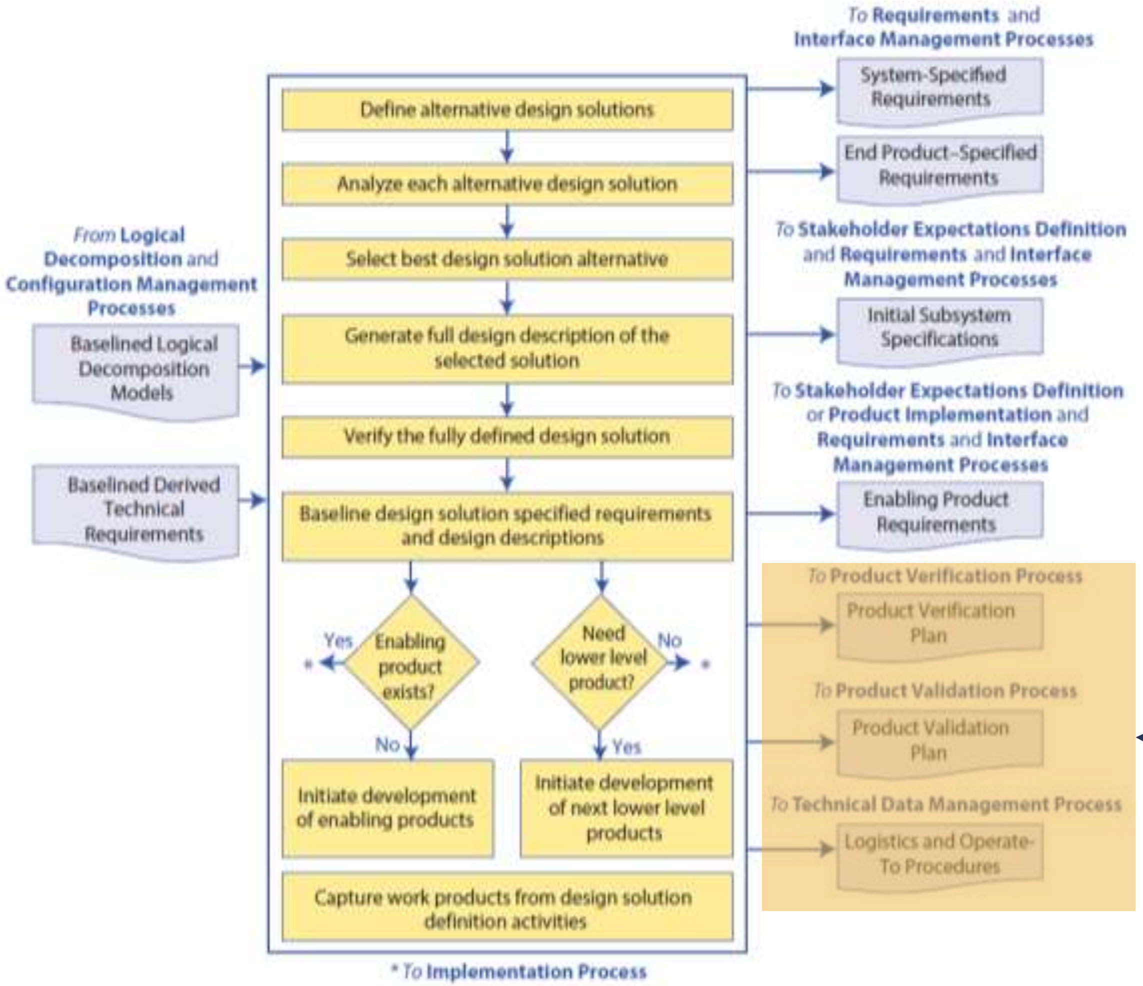
nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

These are not particularly relevant to UNP, but they have some merit:

- Create a feedback loop between design solution options and your Mission Design Document
- Update your System Budgets (e.g. power, data, etc.) as you trade design solution options
- Document the selected design solution (functions, ICD, etc.)

Approved for public release; distribution is unlimited. Public Affairs release approval AFRL-2023-1493

# Design Solution Definition


From Logical Decomposition and Configuration Management Processes
Baselined Logical Decomposition Models
Baselined Derived Technical Requirements
Define alternative design solutions
Analyze each alternative design solution
Select best design solution alternative
Generate full design description of the selected solution
Verify the fully defined design solution
Baseline design solution specified requirements and design descriptions
Yes
Enabling product exists?
No
Need lower level product?
No
Yes
*
*
Initiate development of enabling products
Initiate development of next lower level products
Capture work products from design solution definition activities
* To Implementation Process
To Requirements and Interface Management Processes
System-Specified Requirements
End Product–Specified Requirements
To Stakeholder Expectations Definition and Requirements and Interface Management Processes
Initial Subsystem Specifications
To Stakeholder Expectations Definition or Product Implementation and Requirements and Interface Management Processes
Enabling Product Requirements
To Product Verification Process
Product Verification Plan
To Product Validation Process
Product Validation Plan
To Technical Data Management Process
Logistics and Operate-To Procedures

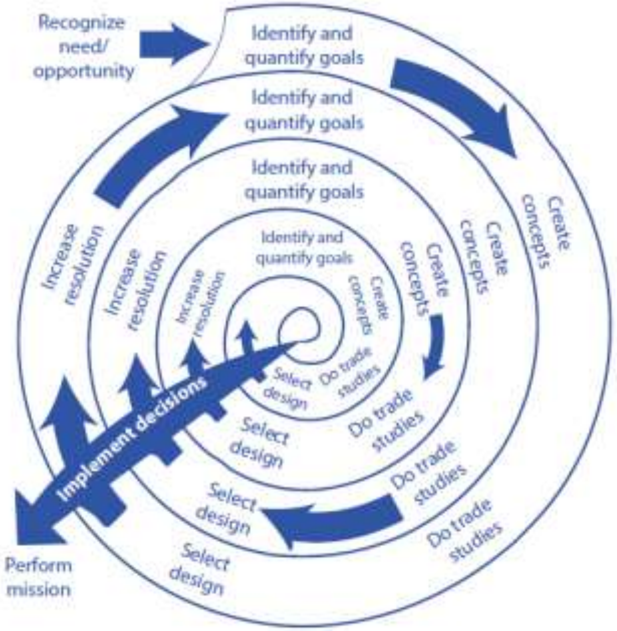
nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

As you start developing design solution definitions, start thinking about verification test procedures, handling procedures, etc.

# Design Solution Definition

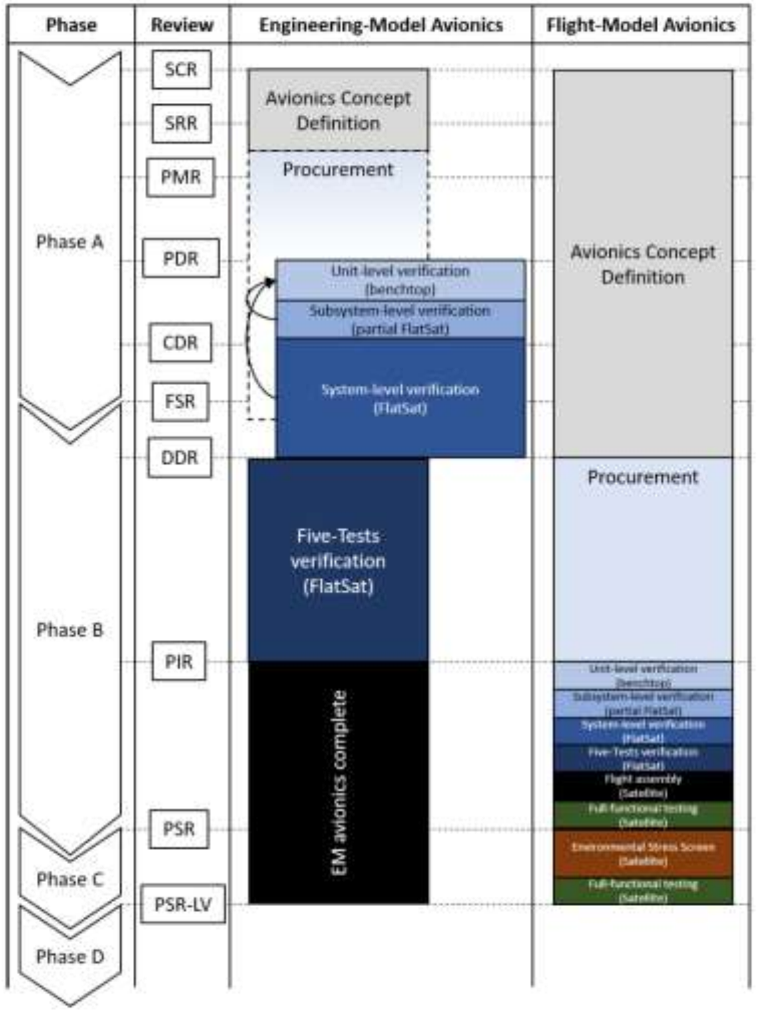
## Design Solution Definitions must be consistent with the CONOPS, Experiment Plan, RVM, & System Budgets

- Requires a successive series of decisions regarding the appropriate design solution option
  - Know your design solution options early, document them, and save them for a future de-scope
  - Always ask: “Why was this solution picked? Is it truly the best option for our team?”
- It’s easy to lose sight of the Mission Design Document, RVM, & System Budgets once you start doing the “fun” stuff (e.g. CAD, PCBs, Code)
  - Always remember to ask the Systems Engineering questions from previous EATs (is this useable? is this the bare minimum required to meet mission success? etc.). The Mission Design Document, RVM, and System Budgets should never “go away” – update them as you go
  - You might find a conflict in your Mission Design Document, RVM, and System Budgets once you start Design Solution Definition/Implementation
    - e.g. there are no reasonably available products capable of providing the functionality you need, the ADCS you need is too expensive, your power budget does not close, etc.
- Make decisions swiftly and confidently, don’t dawdle
  - Do not wait until a formal UNP review to make important design decisions – sometimes there are months between reviews. Make the decision as a team, with your stakeholders, and discuss it at the next formal UNP review. E-mail the UNP PMO office if it is super pertinent. Remember YOU are the responsible design authority for your program



nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf

# UNP Avionics Development



Phase
Review
Engineering-Model Avionics
Flight-Model Avionics
SCR
SRR
PMR
PDR
CDR
FSR
DDR
PIR
PSR
PSR-LV
Phase A
Phase B
Phase C
Phase D
Avionics Concept Definition
Procurement
Unit-level verification (benchtop)
Subsystem-level verification (partial FlatSat)
System-level verification (FlatSat)
Five-Tests verification (FlatSat)
EM avionics complete
Avionics Concept Definition
Procurement
Unit-level verification (benchtop)
Subsystem-level verification (partial FlatSat)
System-level verification (FlatSat)
Five-Tests verification (FlatSat)
Flight assembly (Satellite)
Full-functional testing (Satellite)
Environmental Stress Screen (Satellite)
Full-functional testing (Satellite)

AFRL Image

| Stage | Description | Entrance Criteria | Exit Criteria |
|---|---|---|---|
| Avionics Concept Definition | Avionics Specification in development<br>Conducting trade studies to determine appropriate procurement strategy | Mission-level requirements and their flow down to the subsystem-level complete<br>Baseline Mission Design complete<br>Preliminary system budgets defined (e.g. data, energy, etc.) | Avionics Specification complete<br>Procurement strategy identified |
| Procurement | **In-house**: Team members are designing the unit(s) in-house<br>**COTS**: Vendor(s) identified and quote(s) received. Waiting to receive unit(s) from vendor | Procurement of unit(s) that are intended to fulfill the Avionics Specification has started | Procurement of unit(s) that are intended to fulfill the Avionics Specification is complete |
| Unit-level verification | Teams are conducting unit-level testing to verify unit-level functionality and Avionics Specification fulfillment | The avionics unit is procured and on-hand | Documentation of a successful unit-level verification test complete |
| Subsystem-level verification | Teams are conducting subsystem-level testing to verify subsystem-level functionality and Avionics Specification fulfillment | All units that compose the subsystem are unit-level verified<br>All units that compose the subsystem are integrated with form-fit-function representative interconnects | Documentation of a successful subsystem-level verification test complete |
| System-level verification | Teams are conducting system-level testing to verify system-level functionality and Avionics Specification fulfillment | All subsystems are subsystem-level verified<br>All subsystems are integrated with form-fit-function representative interconnects | Documentation of a successful system-level verification test complete |
| Five-Tests verification | Teams are conducting the Five-Tests with their avionics system | The system has been system-level verified | Five Tests completed (Sec 4.3) and documented (Ch 8) |

# UNP Avionics Development

## Software Development Expectations

| Review | Expectation |
|---|---|
| SCR | Not required |
| SRR | This initial deliverable can be a very short explanation to outline the plans outlined in the software development section of the User's Guide. |
| PMR - PDR | Starting with PMR, the first full draft of this document should be submitted. Everything identified in the document description should be provided in draft form. System level architecture defined (block diagrams, ASM charts, state diagrams, etc.) Updates, as needed, should be provided at PDR. Drivers and firmware necessary for unit-level verification released. |
| CDR - FSR | The document should be fairly mature at this point. Software necessary for subsystem level verification released. Updates should be provided for each review. The submitted document should describe the flight software functionality. Initial release of software intended to fulfill Command Execution Test (CET) and Day-in-the-Life (DitL) test requirements, but may not be fully debugged. |
| Phase B | It is expected that software development may continue to occur in preparation for the Five Tests and as FM hardware is fully integrated and tested. Any changes to the design document supporting that development should be updated accordingly and submitted at each review. |
| Post Phase B | This document should be frozen once shipment to AFRL occurs. |

# COTS Avionics

UNP teams may purchase some of their avionics from manufacturers (i.e. COTS) instead of developing it themselves (i.e. in-house).

A common example is the ADC subsystem: there is a growing market for COTS “black boxes” that contain all the sensors, actuators, support circuitry, and software for common small satellite ADC. COTS procurement approaches appear convenient; especially for teams that do not have expertise in ADC subsystem development.

However, COTS avionics may not always meet technical, cost, or schedule constraints. It is under such circumstances that many UNP teams end up developing PCBs in-house to complete portions of their avionics. A common example is the hardware/software interface between the satellite’s payload and remaining avionics bus. Satellite payloads often require unique power sources (e.g. odd supply voltage or high in-rush current), uncommon data interfaces (e.g. ethernet or custom architecture), and other interface traits that do not directly integrate with typical small satellite COTS solutions. Thus, a custom in-house PCB that provides the appropriate interfaces is often required.

Regardless of the procurement strategy, all avionics must be thoroughly tested to ensure proper functionality and interface compatibility with the remainder of the satellite’s avionics. **It is extremely important to realize that COTS avionics do not guarantee functionality, requirements fulfillment, or interface compatibility with the remainder of the satellite’s avionics.**

# UNP Structural Development

Feb/March 2022

Apr 2022

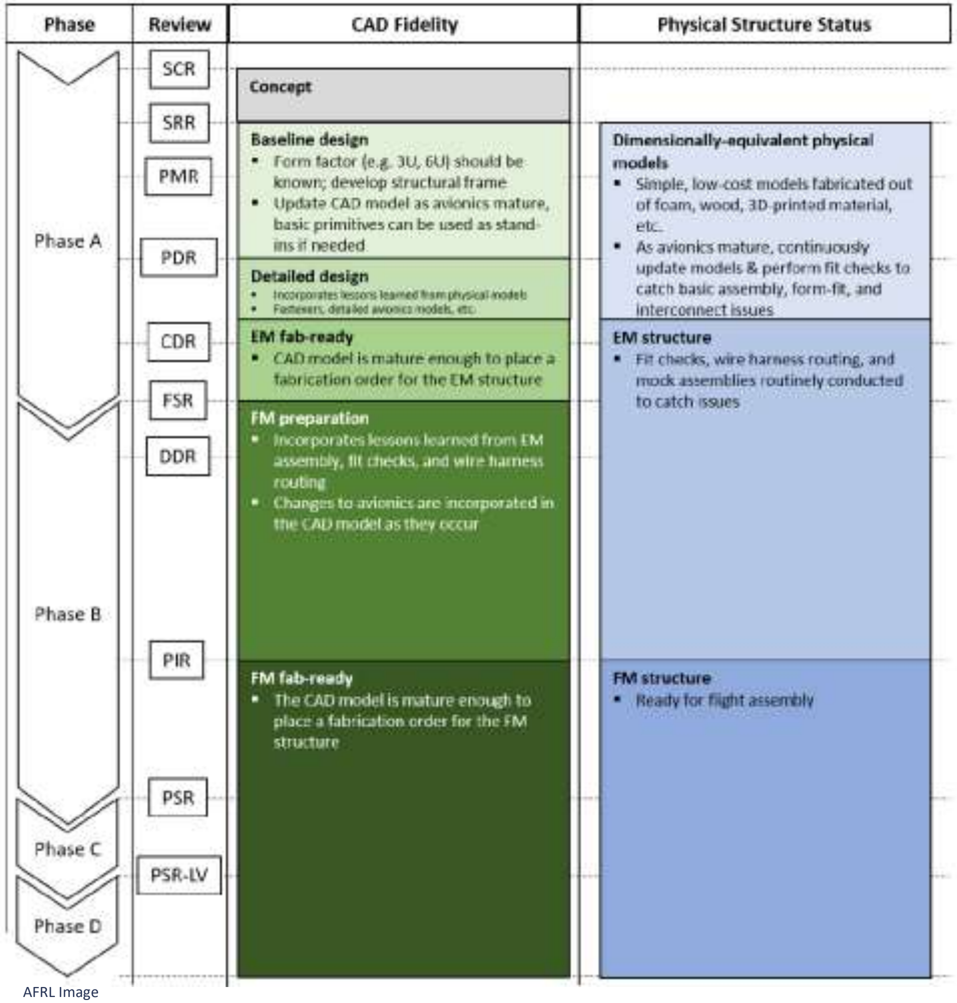
Aug 2022

Nov 2022

Oct 2023

Jan 2024

2024



**CAD Fidelity**

**Concept**

**Baseline design**

- Form factor (e.g. 3U, 6U) should be known; develop structural frame
- Update CAD model as avionics mature, basic primitives can be used as stand-ins if needed

**Detailed design**

- Incorporates lessons learned from physical models
- Fasteners, detailed avionics models, etc.

**EM fab-ready**

- CAD model is mature enough to place a fabrication order for the EM structure

**FM preparation**

- Incorporates lessons learned from EM assembly, fit checks, and wire harness routing
- Changes to avionics are incorporated in the CAD model as they occur

**FM fab-ready**

- The CAD model is mature enough to place a fabrication order for the FM structure

**Physical Structure Status**

**Dimensionally-equivalent physical models**

- Simple, low-cost models fabricated out of foam, wood, 3D-printed material, etc.
- As avionics mature, continuously update models & perform fit checks to catch basic assembly, form-fit, and interconnect issues

**EM structure**

- Fit checks, wire harness routing, and mock assemblies routinely conducted to catch issues

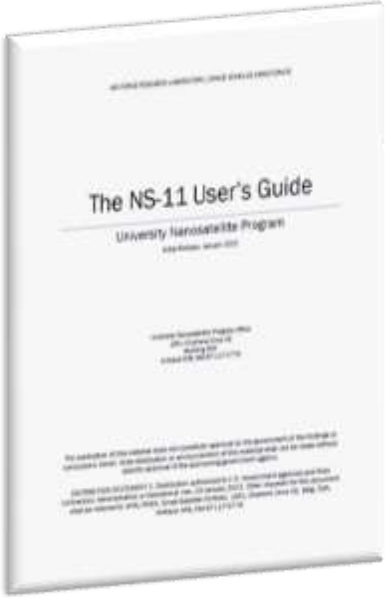
**FM structure**

- Ready for flight assembly

AFRL Image

# All team members should read Chapter 2 of the UNP User's Guide!

# Want to Learn More?



[illegible]

The NS-11 User's Guide

University Nanosatellite Program

[illegible]

**Read the UNP User's Guide!**

**Stick around for future EATs!**

**Ask questions!**

**Read the supporting material**

**Supporting material**

- Most of this presentation copied from “UNP NS10 Systems Engineering Part1 EAT” by Sam Baxendale
- “UNP NS11 User's Guide”, AFRL/RV, 2022
- “Applied Space Systems Engineering”, Larson/Kirkpatrick/Sellers/Thomas/Verma, 2009
- “Space Mission Engineering”, Wertz/Everett/Puschell, 2011
- “The NASA Systems Engineering Handbook”, NASA, 2007
- “INCOSE Systems Engineering Handbook”, INCOSE, 2010
- “Michigan Tech MEPIV Lecture: An Introduction to Systems Engineering”, King, 2019
- “UNP NS9 EAT: Systems Engineering”, Straight, 2016
- “ISO/IEC 15288 IEEE Systems Engineering Standard”, IEEE, 2015
- “ECSS-E-10A European Systems Engineering Standard”, ESA, 2018
- And more! *Not an exhaustive list*

# Questions?