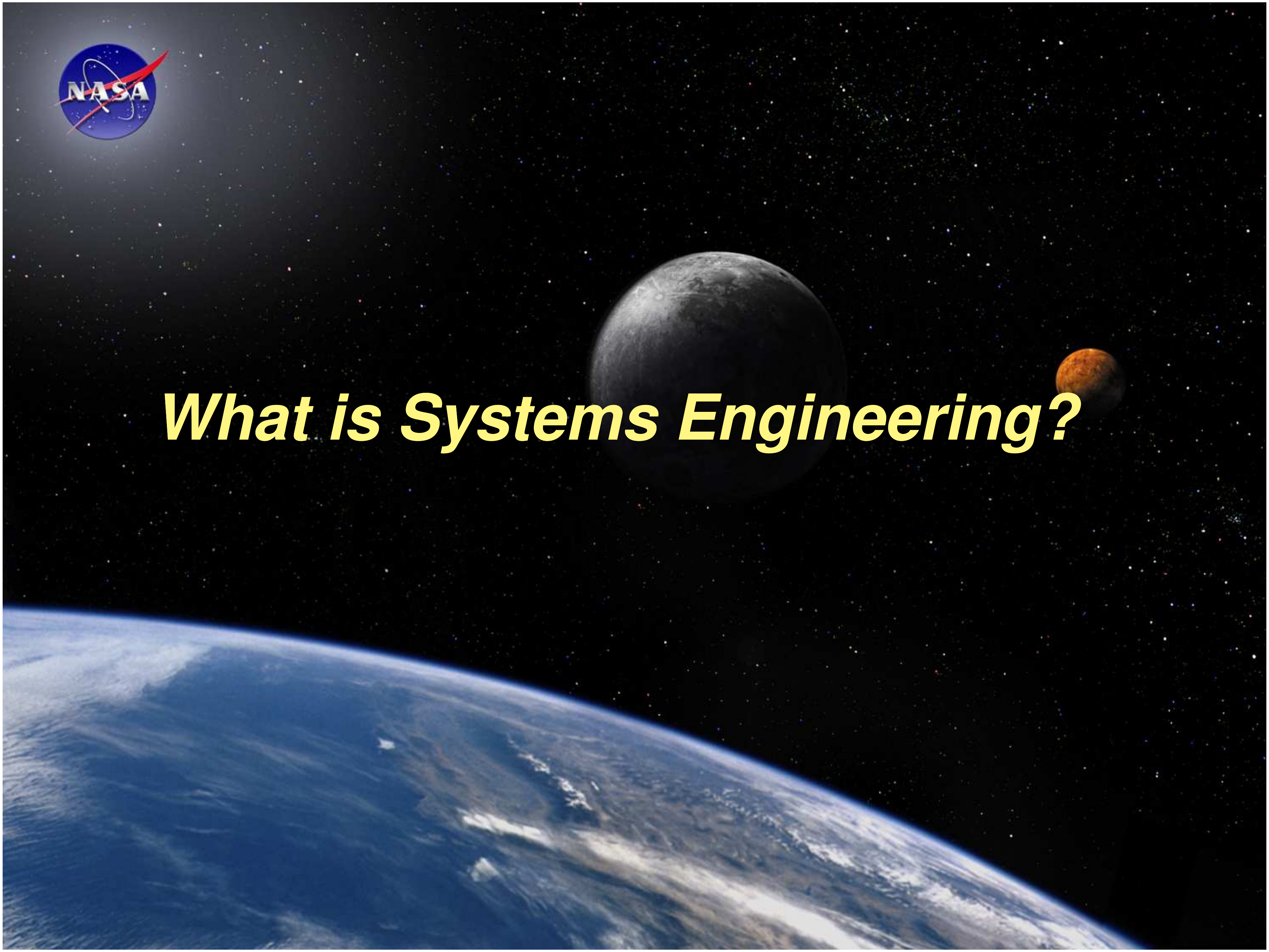

What is Systems Engineering?

What is a System?

Simply stated, a system is an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objectives.

What are examples of a system in the aerospace industry?

Personnel

Facilities

Processes

Hardware

Examples of Systems

- Space Shuttle Main Engine vs. a collection of parts

- Space Shuttle Orbiter with engines and avionics

Examples of Systems

- Space Shuttle Orbiter with solid rocket boosters and external fuel tank
- Space Transportation System (STS) with payload, launch pad, mission controllers, vehicle assembly facilities, trainers and simulators, solid rocket booster rescue ships…

- “System of Systems”
  - STS + International Space Station + TDRSS communication satellites +…

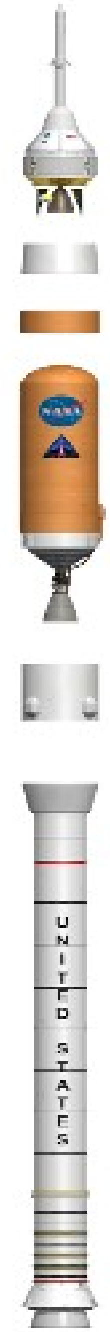
What is Systems Engineering?

Systems engineering **is a robust approach to the design, creation, and operation of systems.**

The approach consists of:

- identification and quantification of system goals
- creation of alternative system *design* concepts
- performance of *design* trades
- selection and implementation of the best *design*
- verification that the *design* is properly built and integrated, and
- assessment of how well the system meets the goals

This approach is iterative, with several increases in the resolution of the system baselines (which contain requirements, design details, verification plans and cost and performance estimates).



Ares 1

System, Systems Engineering, and Project Management

- **System – The combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose.**

- **Systems Engineering – A disciplined approach for the definition, implementation, integration and operation of a system (product or service).**

- **The discipline of systems engineering uses techniques and tools appropriate for use by any engineer with responsibility for designing a system as defined above. That includes subsystems.**

- **Project Management – The process of planning, applying, and controlling the use of funds, personnel, and physical resources to achieve a specific result**

Original Reasons for Systems Engineering

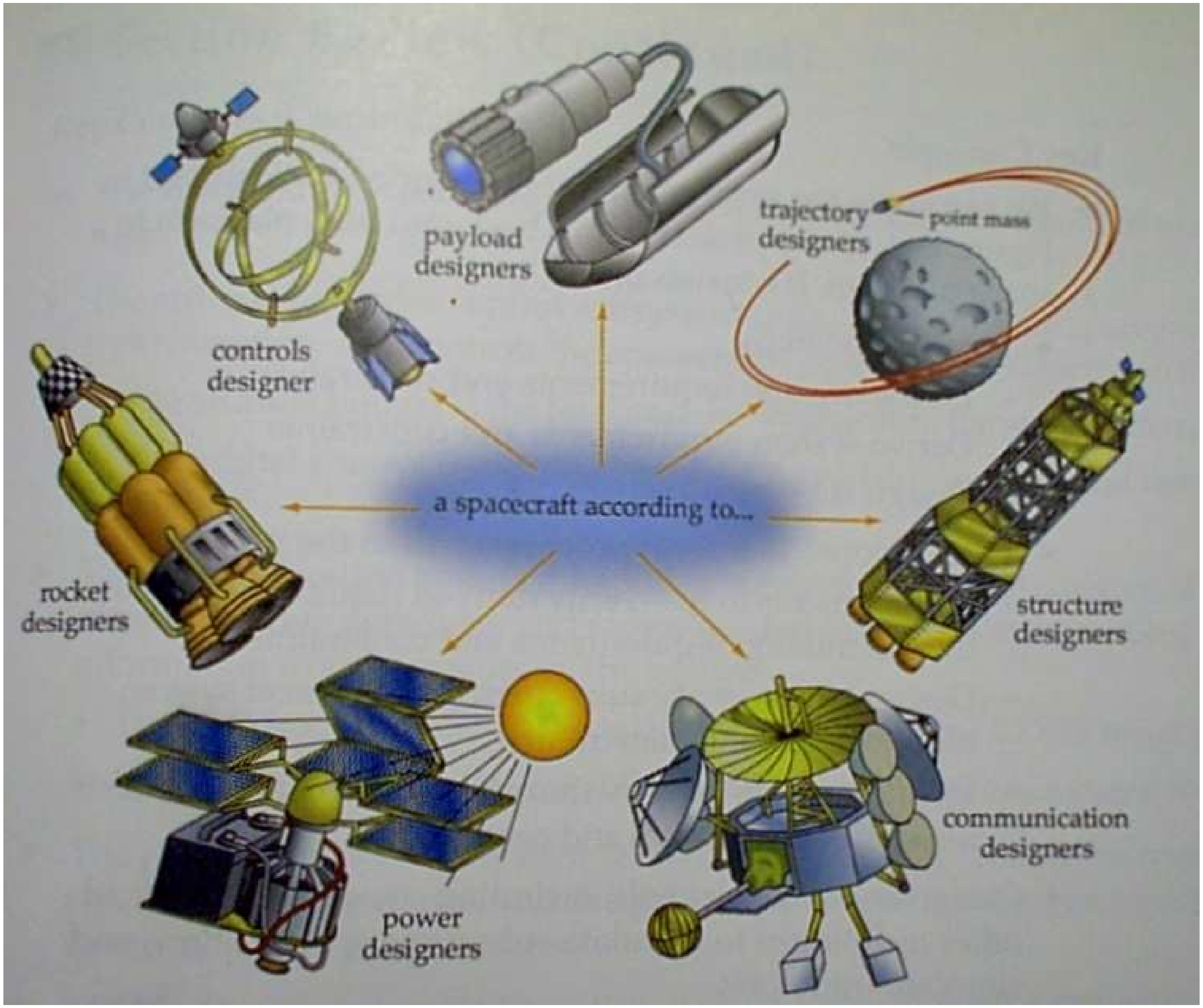
Vasa, Sweden, 1628

Photo from Dec 1999 Civil Engineering magazine

- Systems of pieces built by different subsystem groups did not perform system functions
  - Often broke at the interfaces
- Problems emerged and desired properties did not when subsystems designed independently were integrated
- Managers and chief engineers tended to pay attention to the areas in which they were skilled

- Developed systems were not usable
- Cost overruns, schedule delays, performance problems

Original Reasons for Systems Engineering



- Sometimes individual subsystem designers get so focused on their subsystem designs that they lose sight of the overall mission objectives and requirements

- Good systems engineering coordinates the activities of disciplinary groups with disparate design objectives

Systems Engineering is Built on the Lessons of the Past

- Systems engineering is a relatively new engineering discipline that is rapidly growing as systems get larger and more complex.
- Most of the foundations of systems engineering are built on the lessons of past projects.
- Recurring mission success is codified in techniques and guidelines (e.g., the NASA Systems Engineering Handbook).
- Since mission failures are each unique, their lessons retain their identity.

Declining Systems Engineering Expertise Contributes to a Spectacular Satellite Failure

Future Imagery Architecture - FIA - a $5 billion (award) spy satellite system was behind schedule and expected costs to complete were $13 billion over budget.

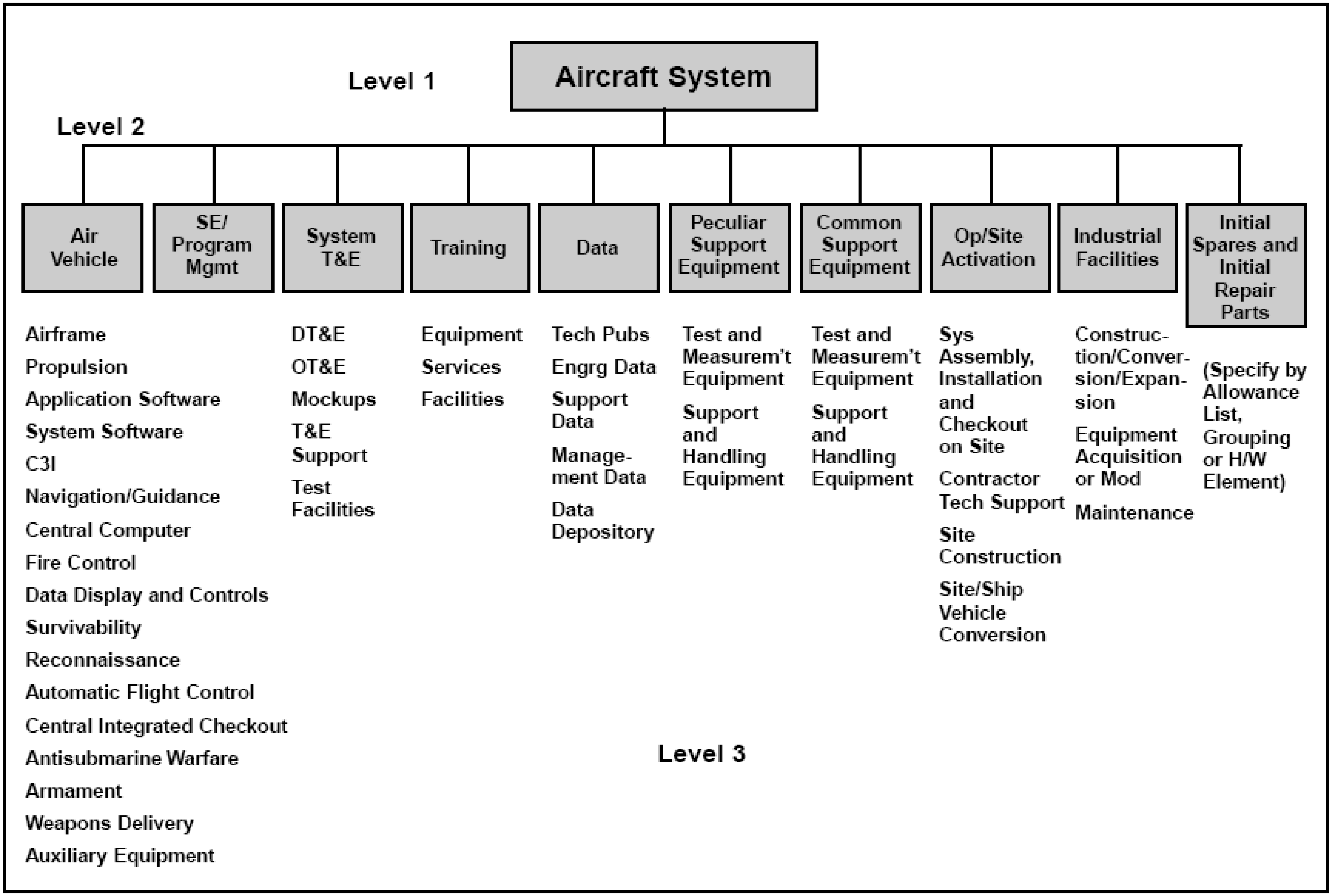
The optical satellite system of FIA was canceled in 2005 after 6 years and spending more than $4 billion.

" … (a) factor was a decline of American expertise in systems engineering, the science and art of managing complex engineering projects to weigh risks, gauge feasibility, test components and ensure that the pieces come together smoothly." NYT, 11/11/07

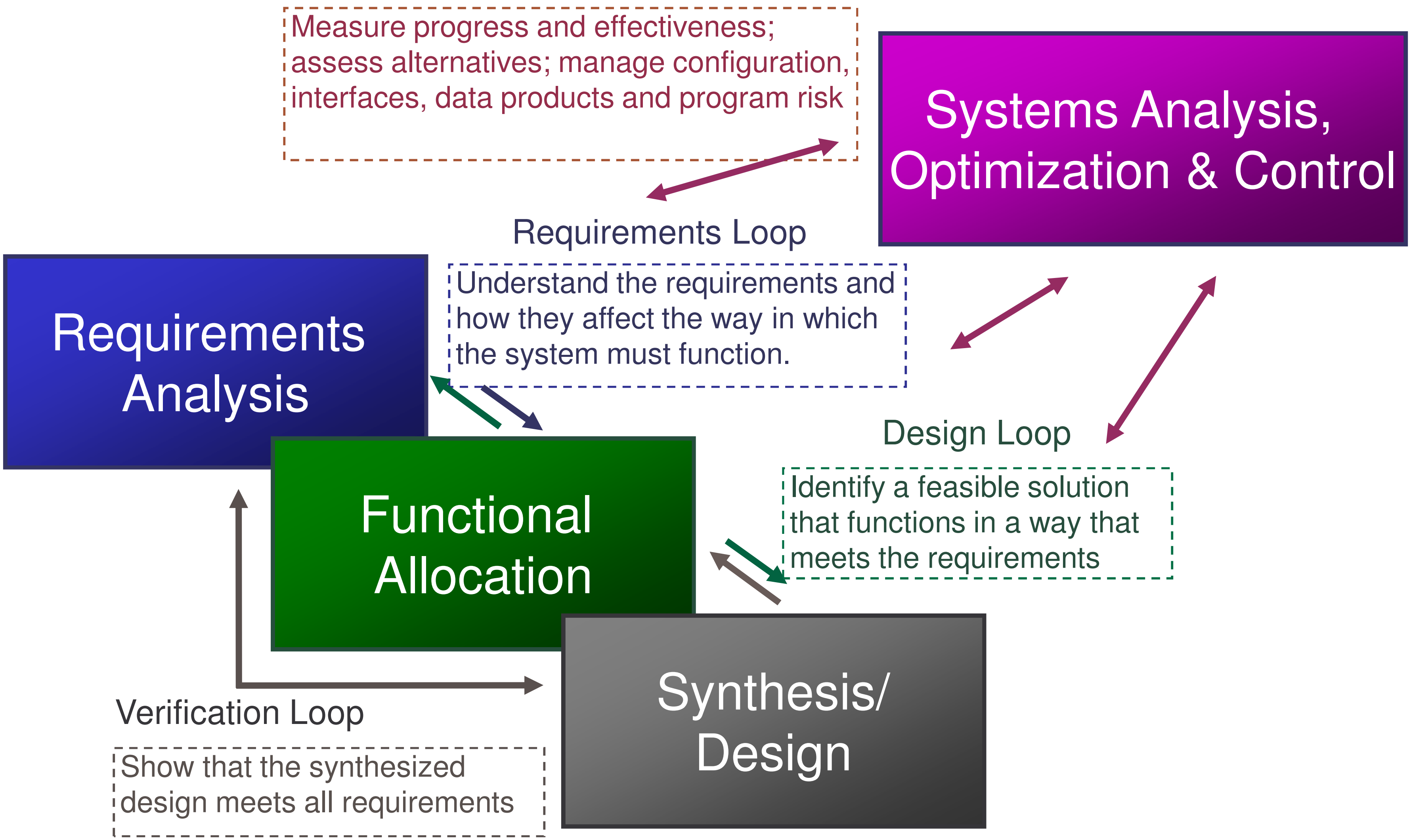
Systems Engineering Process Models Begin with Reductionism

- Reductionism, a fundamental technique of systems engineering, decomposes complex problems into smaller, easier to solve problems - divide and conquer is a success strategy.
- Systems engineering divides complex development projects by <u>product and phase</u>.
- Decomposing a product creates a hierarchy of progressively smaller pieces; e.g.,
  - System, Segment, Element, Subsystem, Assembly, Subassembly, Part
- Decomposing the development life of a new project creates a sequence of defined activities; e.g.,
  - Need, Specify, Decompose, Design, Integrate, Verify, Operate, Dispose

Subsystem Organization Structure



Systems Engineering
Process also Includes Requirements Analysis



With Process Comes Systems Engineering Practices

Set up a plan for each of these ***EARLY!***

Design Budgets
- Power
- Memory/data
- Communications
- Mass
- $$$
- Other resources

Documentation Organization
- Requirements (**!!**)
- Materials Lists
- CAD drawings
- Safety documents
- Interface controls
- Configuration management

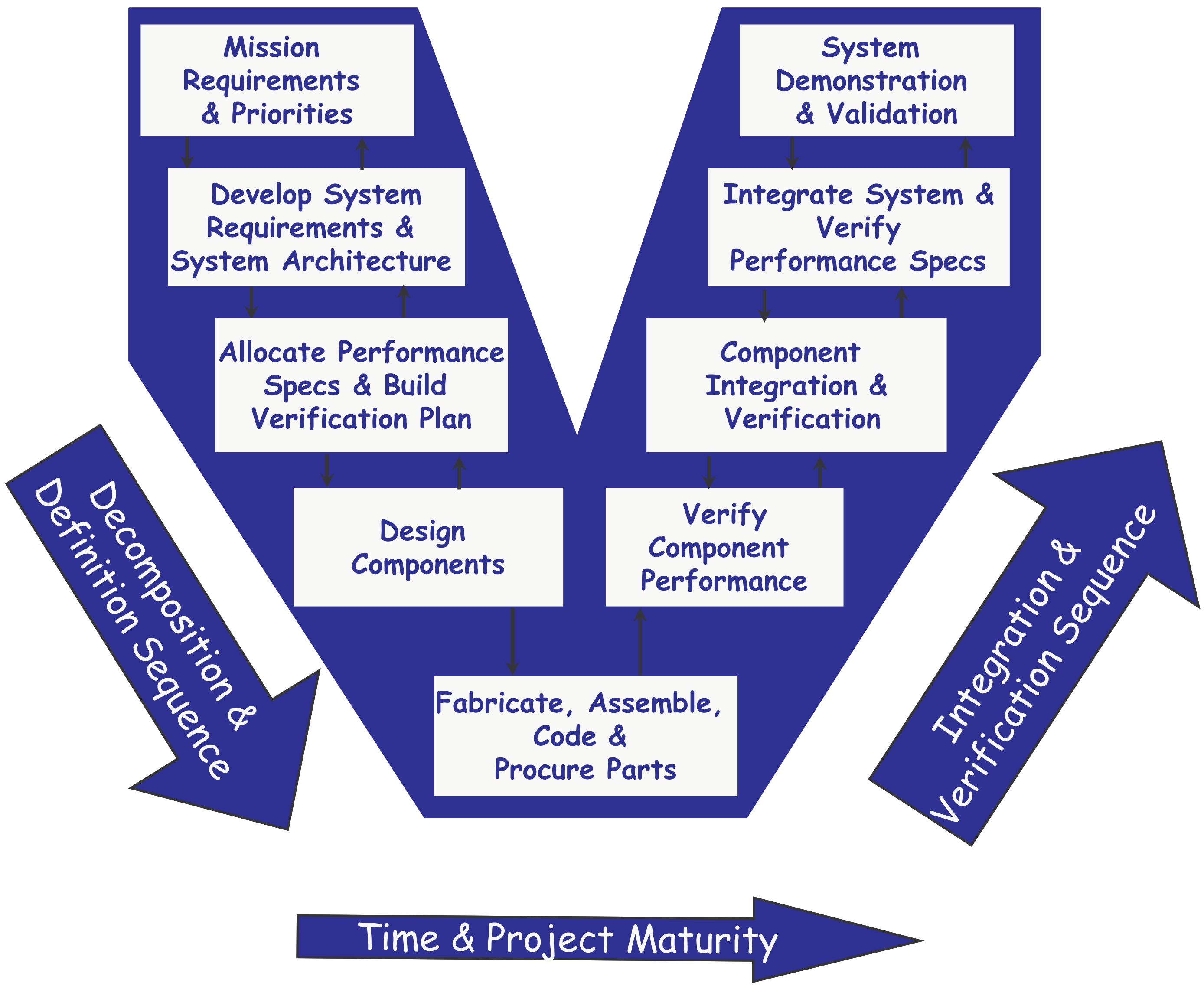
Acquisition strategies
- Purchase
- In-house
- Contribution
- Other

Interface Control
- Harness & Connectors
- Structural connections
- Software protocols & signal processing

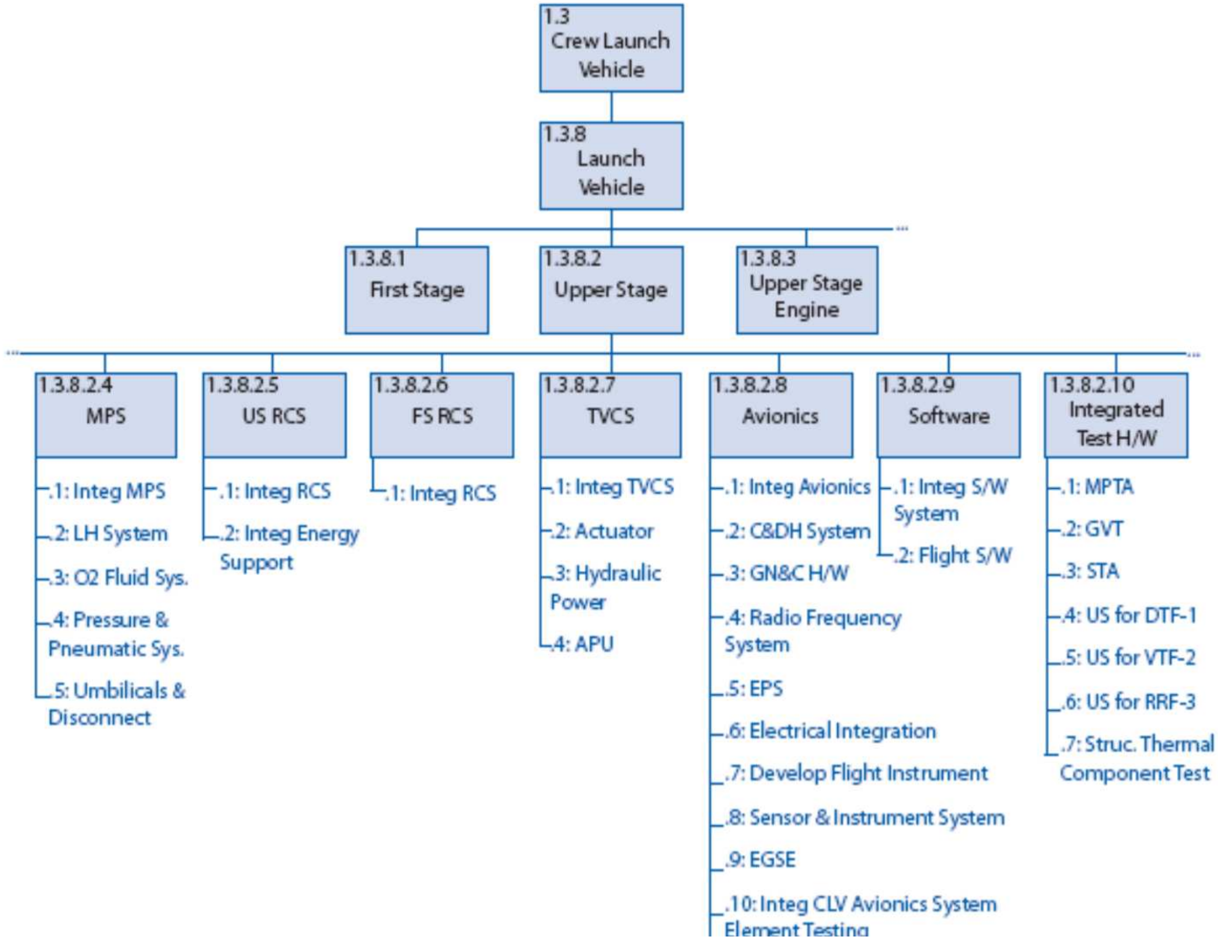
Identify design drivers
- Cost
- Schedule
- Performance

Execute a risk management plan

The Systems Engineering 'Vee' Model Extends the Traditional View with Explicit Decomposition and Integration

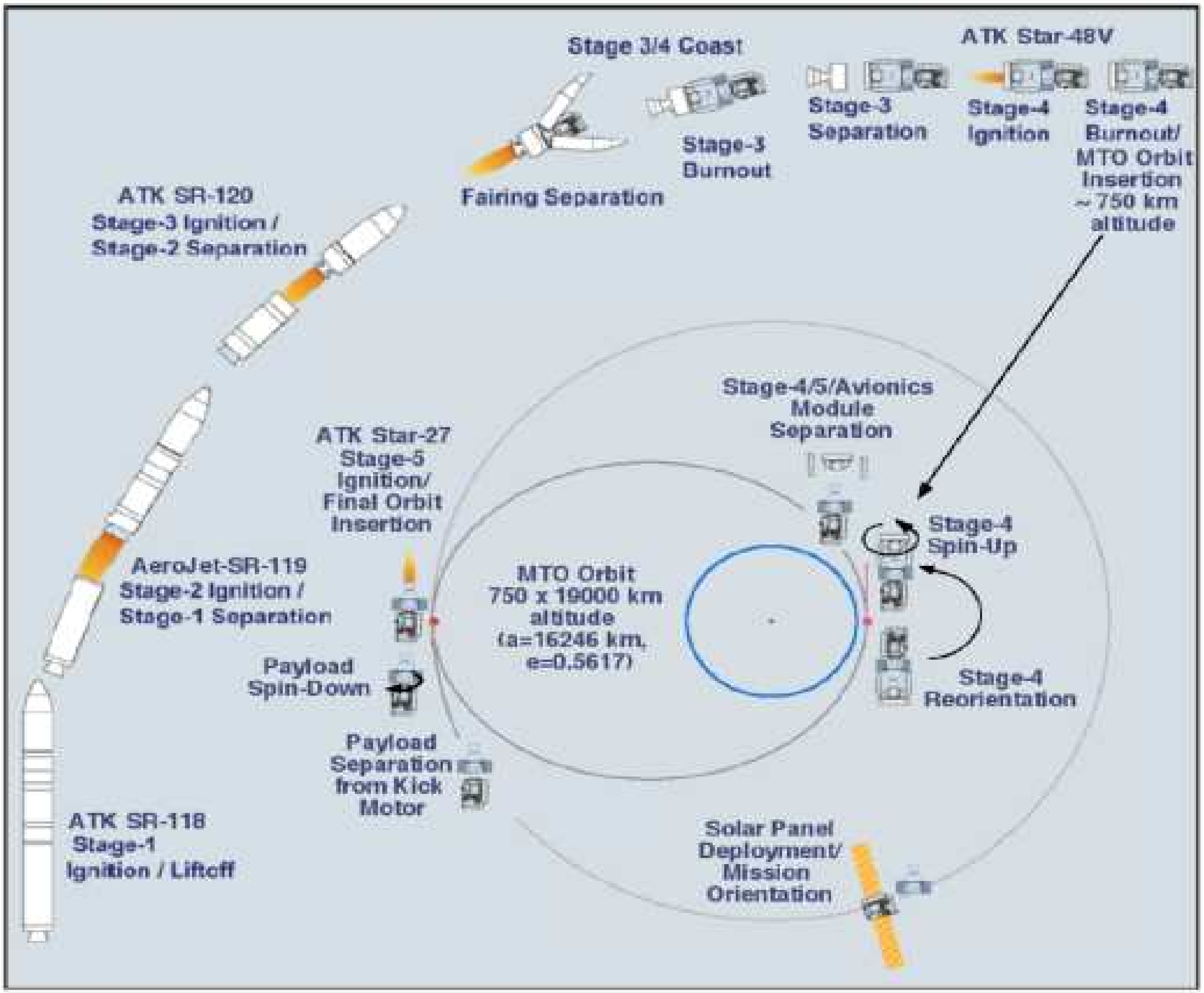


Production and Work Breakdown Structure



Allows the systems engineer to systematically divide an entire project into a set of major production areas including, sub-areas, and sub-sub areas.

Concept of Operation (Capabilities)



Launch and Deployment Concept of Operations.

Short Verbal or graphic statement, in broad outline, of a customer's assumptions or intent in regard to an operation or series of operations.

Trade Studies

- Trade study is a tool used to help choose the best solution among alternatives.
- Numerical values are given based on weight factors and a normalization scale for the evaluation criteria.

Decision Factors / Alternatives	Range Wt. = 2.0		Speed Wt. = 1.0		Payload Wt. = 2.5		Weighted Total
	U	W	U	W	U	W	
Transport System 1	.8	1.6	.7	.7	.6	1.5	3.8
Transport System 2	.7	1.4	.9	.9	.4	1.0	3.3
Transport System 3	.6	1.2	.7	.7	.8	2.0	3.9
Transport System 4	.5	1.0	.5	.5	.9	2.25	3.75

Key: U = Utility value
W = Weighted value

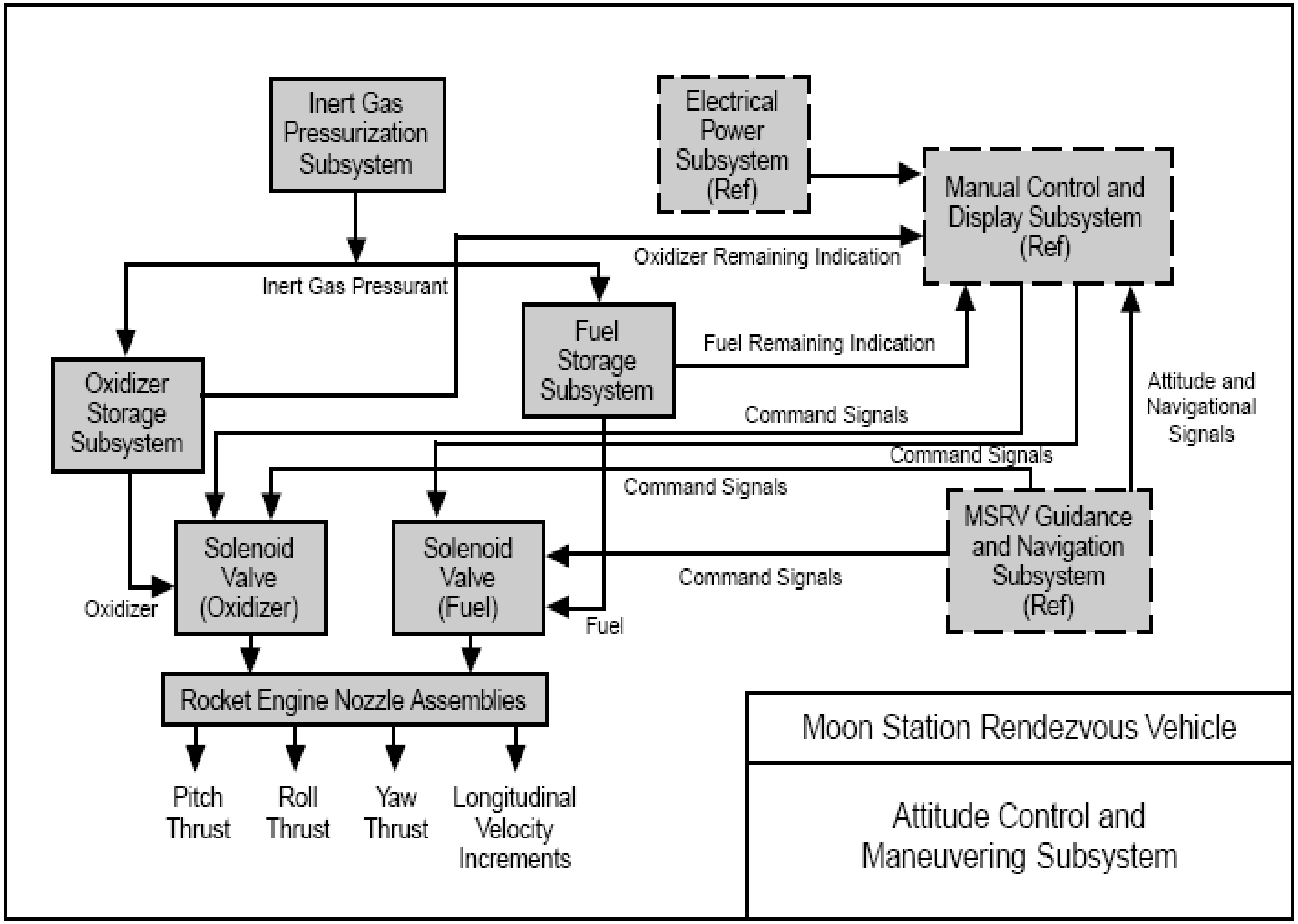
Trade Studies

Study Example – Comparison of Controllers for CubeSat

Microcontroller vs. FPGA Trade Study

MCU vs. Antifuse FPGA Trade Study V1.0					
Criteria	**Weight (%)**	**Microcontroller**	**Grade**	**Antifuse FPGA**	**Grade**
Radiation Tolerance	30%	Logical	2	Physical (rad hard by design)	5
Programming Language	20%	C	4	VHDL or Verilog	2
Power consumption	15%	16.5 mW	4	<16.5 mW	5
Cost per unit	10%	$15.05	4	$30	2
Initial Cost	5%	$0.00	5	$500	2
In Flight Programmable	5%	Yes	5	No	1
CubeSat Legacy	15%	Extensive	3	Unknown	1
Average Score			**3.8571**		**2.57143**
Weighted Score			**3.35**		**3.15**

Functional Block Diagrams



Schematic Block Diagram Example

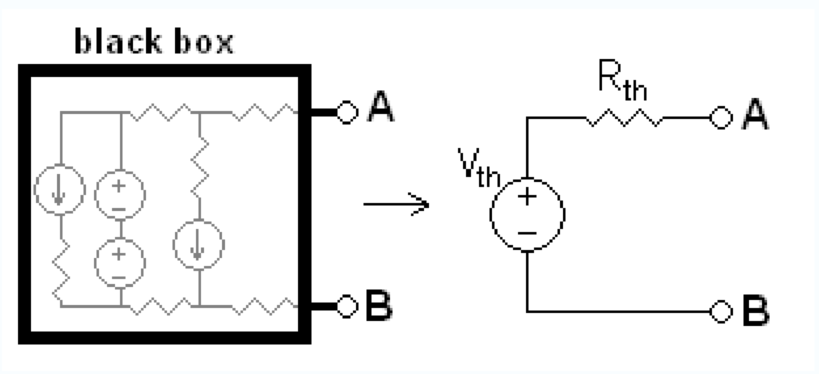
Schematic Block Diagram (SBD) depicts hardware and software components and their inter-relationships.

Developed at successively lower levels as analysis proceeds to define lower-level functions within higher-level requirements.

Useful for developing Interface Control Documents *(ICD's)*

Interface Control Document

-- Allows Disparate groups to work integrate sub-systems without complete working knowledge of what is inside of the “black box
-- In this way, independent teams can develop the connecting systems which use the interface specified, without regard to how other systems will react to data and signals which are sent over the interface.
-- An adequately defined ICD will allow one team to test its implementation of the interface by simulating the opposing side with a simple communications simulator.



Interface Control Document

ANALOG SIGNAL NAME(INTER-MODULE)	WHAT SYSTEM	HOW COMMUNICATED TO C&DH	Voltage Range	PIN #'s	CONNECTOR	COMMENT
Power Voltages						
Gnd	EPS	NA	0V			signal ground
3.3V regulated supply	EPS	CDH MCU ADC (PF0)	3.3V			C&DH power
5.0V regulated Supply	EPS	CDH MCU ADC (PF1)	5V			TNC power, XCVR power
unregulated supply	EPS	CDH MCU ADC (PF2)	3.3-4.25			nothing directly uses
EPS Voltages						
bat1	EPS	via I2C from ADC1 on EPS	3.7V	I2C		
bat2	EPS	via I2C from ADC1 on EPS	3.7V	I2C		
solar cell output	EPS	via I2C from ADC1 on EPS	2.5V	I2C		the entire array
unregulated supply	EPS	via I2C from ADC1 on EPS	3.3-4.25V	I2C		nothing directly powered from this
EPS Currents (sent as voltage)						
bat1 charging current	EPS	via I2C from ADC3 on EPS		I2C		
bat 1 discharging current	EPS	via I2C from ADC3 on EPS		I2C		
bat2 charging current	EPS	via I2C from ADC3 on EPS		I2C		
bat2 discharging current	EPS	via I2C from ADC3 on		I2C		

Interface Control Document

TNC Interface V1.0				
Pin			Description	MCU Pin
1	CTS	Clear to Send	RS-232 level flow control signal out of the TNC. Indicates whether the TNC is allowing or holding off data input on pin 3.	PE4
2	RXD	Receive Data	RS-232 level data out of the TNC.	PE0
3	TXD	Transmit Data	RS-232 level data into the TNC.	PE1
4	RTS	Request to Send	RS-232 level flow control signal into the TNC. Indicates the MCU wants to send data to the TNC.	PE3
5	GND	Ground	Common signal and frame ground.	GND
6	DCD		No connection.	N/C

Table 4: MCU/TNC Interface

Power and Mass Budget Analysis

C&DH Mass Budget V1.0			
Part	**Mass (kg)**	**Quantity**	**Mass Total**
Memory	0.0030	1	0.0030
ATmega2561L	0.0010	2	0.0020
3V relays	0.0003	4	0.0012
I2C ADC's	0.0010	3	0.0030
I2C GPIO	0.0010	1	0.0010
Circuit Board	0.0250	1	0.0250
Crystal	0.0010	1	0.0010
Miscellaneous	0.0100	1	0.0100
Thermistors	0.0010	8	0.0080
Total	**0.0433**		**0.0542**
Contingency			**10%**
Total Plus Cont.			**0.0596**

Table 5: C&DH Mass Budget

Weight and power growth are major enemies of many products.

Power and mass budget analyses insure product growth is bounded and eventually mandates comes in "under weight" and "overpowered"

Power and Mass Budget Analysis

C&DH Power Budget

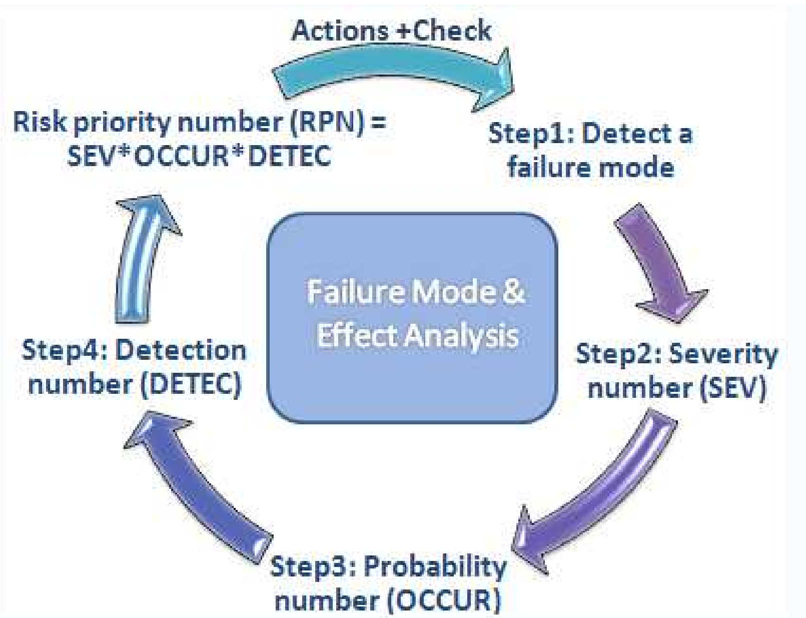
C&DH Power Budget V1.0											
		Voltage Range			Current	Power Max	Power Mode			
Part	Quantity	Min	Max	V used	mA	mW	Safe	Idle	Normal	Transmit
Memory	1	2.7000	3.6000	3.0000	4.0000	12.0000	0.1000	0.1000	1.0000	1.0000
ATmega2561L	2	1.7000	5.5000	3.0000	5.5000	33.0000	16.5000	16.5000	16.5000	16.5000
3V relays	4	1.0000	5.0000	3.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
I2C ADC's	3	2.7000	5.0000	3.0000	0.2250	2.0250	1.0125	1.0125	2.2050	2.2050
I2C GPIO	1	2.3000	5.5000	3.0000	0.1040	0.3120	1.0000	1.0000	1.0000	1.0000
Circuit Board	1	~	~	~	~	0.0000	0.0000	0.0000	0.0000	0.0000
Crystal	1	~	~	~	~	0.0000	0.0000	0.0000	0.0000	0.0000
Miscellaneous	1	~	~	~	~	0.0000	0.0000	0.0000	0.0000	0.0000
Thermistors	8	1.0000	5.5000	3.0000	0.3333	7.9992	8.0000	8.0000	8.0000	8.0000
Total					**10.1623**	**55.3362**	**26.6125**	**26.6125**	**28.7050**	**28.7050**
Contingency						**15%**	**15%**	**15%**	**15%**	**15%**
Total Plus Cont.						**63.6366**	**30.6044**	**30.6044**	**33.0108**	**33.0108**

Table 6: C&DH Power Budget

Failure Modes and Effects Analysis

-- A failure modes and effects analysis (FMEA) is a procedure for analysis of potential failure modes within a system for classification by severity or determination of the effect of failures on the system.

--FMEA provides an analytical approach, when dealing with potential failure modes and their associated causes.



Failure mode: The manner by which a failure is observed; it generally describes the way the failure occurs."

Failure effect: Immediate consequences of a failure on operation, function or functionality, or status of some item

Module Summary: What is Systems Engineering?

- *Systems engineering* is a robust approach to the design, creation, and operation of systems.
- Systems engineering is a ubiquitous and necessary part of the development of every space project.
- The function of systems engineering is to guide the engineering of complex systems.
- Most space projects struggle keeping to their cost and schedule plans. Systems engineering helps reduce these risks.
- Systems engineering decomposes projects in both the product and time domain, making smaller problems that are easier to solve.
- System decomposition and subsequent system integration are foundations of the Vee and the NASA systems engineering process models.

Eight Rules for Prototyping *(1)*

1 Recognize That Ideas Are Cheap – Given the connected, Internet-savvy world in which we live, ideas have become cheap and they will probably become cheaper with time. The expense lies in testing and verifying what has economic value. A great prototype is often the best way to start a dialogue with potential customers and test your idea's value.

2 Start with a Paper Design – You may be eager to start coding or designing the electronics too quickly. Fight the urge. Writing code without real consideration for several design factors leads to heartache and a lot of rework. Start with a simple paper design. For a user interface or Web software prototype, a paper design is efficient and effective for quickly working through the functionality. You can get peers and, hopefully, customers to give feedback on where images, text, buttons, graphs, menus, or pull-down selections are located. Paper designs are inexpensive and more valuable than words.

Eight Rules for Prototyping *(2)*

3 Put in Just Enough Work – Know your objectives and stick to them. There are two good reasons to prototype: the first is to test the feasibility of a hardware or software architecture, and the second is to create a demonstration and gain customer feedback so you can price and put a value on your innovation. Keep these objectives in mind and be careful not to fall in love with the process. Prototyping is fun and innovators love to tinker, but you want to invest just enough time and work to meet the objectives.

4 Anticipate for Multiple Options – Design your prototype with modularity in mind. Great prototypes are often modular, which means you can quickly adapt them to meet customers' unforeseen needs. Customers ultimately decide how to use your product, not you. Design in options for expansion, performance, packaging, and lower cost.

Eight Rules for Prototyping *(3)*

5 Design for Reuse in the Final Product – The ideal situation is to design a prototype you can produce and distribute in high volume. Not many prototyping tools can deliver on this promise. Typically you give up performance for design flexibility. Look for prototyping tools that make it possible for you to scale your prototype from lab to market.

6 Avoid Focusing on Cost Too Early – For hardware designs, a potential time sink and pitfall is getting caught up in endless cost optimization analysis during the early stages of your prototype design. Cost is always important, but your goal with a prototype is to be within striking distance of a profitable design. Initially, focus on proving the value of your innovation, and design with modularity in mind. While frustrating, your design may follow many paths that do not ultimately lead to value. Focus on securing your first set of customers and then work on cost optimization.

Eight Rules for Prototyping *(4)*

7 Fight "Reversion to the Mean" – When prototyping, the tendency is to develop something easy rather than develop something that has a "wow" factor. Stay true to your vision and make sure your prototype captures the original thought of your innovation.

8 Ensure You Can Demonstrate Your Prototype – Your prototype should be easy to demonstrate. With customers, venture capitalists (VCs), and potential employees, you want to start strong and show the most amazing capabilities first. Do not build up to a crescendo. Most people's attention spans are limited to less than 60 seconds. In presentations, whether they are for a new employee or a VC, get to the demonstration as fast as possible. If the demonstration is amazing, all else falls into place.

http://zone.ni.com/devzone/cda/pub/p/id/579?metc=mtnxdy

Keys to Holding a Successful Meeting *(1)*

• Meetings are essential to any team effort, be it designing a rocket System, or launching a new cosmetic product

• Done properly, meetings can quickly disseminate information, solve problems, create consensus, and get everyone "on the same page"

• Done improperly, meetings can bog down, cause dissention, delay, and sometimes cripple a project.

• Every meeting must a specific purpose – before arranging a meeting one need to think precisely about what it is that needs to be accomplished.

Keys to Holding a Successful Meeting *(2)*

- **Typical Meeting Purposes"**
  - **Brainstorming new ideas**
  - **Developing an idea or plan**
  - **Having a progress update**
  - **Technical interchange**
  - **Considering options and making a collective decision**
  - **Selling something to a potential buyer**
  - **Building a relationship with somebody**

There may be a mixture of objectives and desired outcomes for a particular meeting,; however, primary objectives should kept clearly in mind and those should prioritiszed above others.

Keys to Holding a Successful Meeting (3)

1. **Invite the right people. Make sure these people attend.**

2. **Start with a clear objective for the meeting. Particularly with routine meetings, it's tempting to hold the meeting because it's "checking a box", but what are you really trying to accomplish? People don't actually bond very much in unproductive meetings that lack clear objectives.**

3. **Set up a written agenda in advance. As you build the agenda, get real about how long it will take to address each topic. As a guideline, assume that if the goal is to make a decision, it will take four times longer than if the goal is to simply provide a status report.**

Keys to Holding a Successful Meeting *(4)*

4. **Formally track problem-solving and decision-making discussions. If everyone is in same room, use a flipchart or whiteboard, otherwise use electronic recording media. Appoint someone to take notes at the beginning of the meeting. Formally archive meeting notes in a data base with access to participating team members.**

5. **Formal Tracking Tools:**
   - ***a. Action Items – Requests for Action (RFA)***
     - *Who is assigned action?*
     - *When is action due?*
     - *Who are action's "customers"*
   - ***b. Information Items – Requests for Information (RFI)***
     - *Who provided the information and verification?*
     - *When is action due?*
     - *Who needs the information*