Systems Engineering
Fernando Pellerano
NASA Goddard Space Flight Center
What is Systems Engineering?

• What’s a System?: The combination of elements that function together to produce the capability to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose.

• Why does everyone has a different idea of what’s Systems Engineering?
• Systems Engineering involves breaking a complex problem up into smaller, more manageable pieces.
• Systems Engineering functions (Sarah A. Sheard, 1996 INCOSE Symposium):
  – Requirements Owner
  – System Designer
  – System Analyst
  – Validation/Verification Engr.
  – Logistics/Ops Engineer
  – Glue Among Subsystems
  – Customer Interface
  – Technical Manager
  – Information Manager
  – Process Engineer
  – Coordinator
  – Classified Ads SE
What’s the Motivation?

NASA’s current paradox is cost control while achieving technical performance.

Percent of Project Cost Spent for Requirements Development

1. Higher-level Needs, Objectives & Constraints

2. Assessment of functions and operations

3. Iterative Process

4. Architecture/Design Concepts

5. Prediction

6. Converged & Viable Solution

The "final" touch

Validation

Requirements Documentation

Requirements Management

TFAWS 2015 – August 3-7, 2015
Requirements

• Use the word “shall” when defining requirements and only for requirements:
  – “The attitude control system shall point the instrument boresight to within 6 arc seconds of the commanded attitude.”
  – Make it clear to the reader exactly what must be done.

• Requirements should be:
  – Measurable
    • Avoid vague statements like “shall point as accurately as possible”
  – Verifiable
    • If no one can demonstrate that a system does or does not meet a requirement, then there is no requirement

• Document rationale!
  – Capture the why while it is fresh.
  – Enable future changes—people are afraid to change things they don’t understand.
  – Challenge “borrowed” or legacy requirements – “we’ve always done it that way” is not a justification
Requirement Examples (1)

• Don’t use ambiguous terms
  – The ATLAS laser shall generate laser pulses at approximately 10kHz repetition frequency.
  – *The ATLAS laser shall generate laser pulses at a 10kHz +/- 0.3 kHz repetition frequency.*

• Positively stated
  – The transmitter shall not allow radiation of the second harmonic to be greater than 20dB.
  – *The transmitter shall suppress radiation of the second harmonic by greater or equal to 20dB.*

• Keep one thought
  – The ATLAS laser shall generate laser pulses with a center wavelength of 532 nm +/- 15nm at 10kHz +/- .3kHz repetition frequency.
  – *The ATLAS laser shall emit light at 532nm nm +/- 15nm.*
  – *The ATLAS laser shall generate laser pulses at a 10kHz +/- 0.3 kHz repetition frequency.*
• Use consistent terminology
  – The ATLAS transmitter shall generate laser pulses at a 10kHz +/- 300Hz repetition frequency.
  – The ATLAS laser shall generate laser pulses at a 10kHz +/- 0.3kHz repetition frequency.

• Avoid “people do” statements
  – The operator shall send stored data to the science team.
  – The Ground System shall transmit stored data to the science database upon user request.

• Avoid implementation specific
  – The ATLAS laser shall use an intermittent signal disruption to generate laser pulses at a 10kHz +/- 0.3 kHz repetition frequency.
  – The ATLAS laser shall generate laser pulses at a 10kHz +/- 0.3 kHz repetition frequency.
Optimal Number of Requirements

- The Quantity & Quality (level of detail) of requirements evolve with time.

  - Too Few or too late requirements under constrains the design solutions.
  - Progress, but not focused resulting in the design not converging to an implementation.

  - Too Many or too early requirements over constrains the design solutions.
  - Progress choked because any decision results in a violation of a requirements.

- If you cannot recall from memory your requirements, you probably have too many.

*We have an obligation to close paperwork of each and every requirement, don’t get carried away and needlessly create busy work for yourself!*
Don’t Let This Happen To You

In order to control cost and schedule, we need to discipline ourselves to do the upfront work before getting into the fun technical work.
# Elements of Space Systems Design

- Orbital Dynamics
- Structures and Mechanisms
- Attitude Control
- Electrical Power
- Thermal
- Command and Data Handling
- Propulsion
- Communications
- Information Systems/Software
- Space Environment
- Launch Vehicle

- Instruments/Payloads
- Reliability
- Safety
- Operations
- Disposal
- Project management
- Maintenance
- Storage
- Verification (system built right?)
- Validation (right system built?)
- …
How Other Disciplines Interface with Thermal

**Mechanical**
- Selection of Materials
- Mounting Design
- Thermal Hardware Locations
- Radiator Accommodation
- TVAC (Deployment, Mechanisms)
- Surface Coating Specification
  - STOP Analysis
  - Geometry Files for Modeling
    - Integration

**Electrical/Power**
- Power Resources (Htrs, TECs, etc)
- Thermal Dissipations
- Grounding/ESD Requirements
  - Voltage Requirements
  - Thermal Sensor Capability
    - Heaters Services
  - TVAC (PDU, Solar Arrays, etc)
    - Integration

**GNC**
- Orbit Requirements
- Pointing Requirements
- TVAC (Reaction Wheels, etc)

**Software**
- Monitoring and Limits
- Temperature Control
- Autonomous Commands

**Contamination/Coatings**
- Material and Coatings Selection
- Temp Predicts & Geometry Model for Contamination Analysis
  - Properties for Thermal Analysis
  - Coating and Tape Application
    - TVAC Requirements

**Optical**
- Predicts for STOP/Pointing
- Temp Control for Stability & Focus

Your Systems Engineering team should serve as a bridge or facilitator among all Disciplines.
Be Aware of Your Stakeholders

- Observatory Manager
- Designers
- Mission Systems Engineer
- S&MA
- Scientists
- Project Manager
- Interfacing Subsystem PDLs
- Spacecraft SE
- Branch Management
- Instrument SE
- Operators
- Technicians
- Manufacturing
- Testers
Observation Strategy & Operations Concept

• Thermal plays a very important role in these early phases of the mission

• How the system will be used in many respects drives the architecture
  – Earth orbit selection given instruments’ observations
  – Planetary, e.g., extreme hots or extreme colds
  – Many instruments’ performance depends on cold temperatures and/or tight stability

• Many requirements will fall out of the ops concept.
  – Temperature ranges and stability
  – Modes of operations over different phases
    • Launch operations
    • Orbit maneuvers
    • Calibration scans
    • Ground testing, e.g., heat pipe orientations
• Example:
  – Aquarius/SAC-D orbit selection
    • Instrument needed to observed the night side “all the time”
    • Room temperature detectors but <100mK stability, >7-days
    • Selection of sun-synch 6am orbit, very stable and ample power
  – ICESat-2 orbit selection
    • Instrument wants to map as much of the poles as possible AND provide high sampling
    • Optics benefit from good thermal control (not to tight)
    • Orbit selected provides a highly variable thermal environment, tough job!
    • Instrument thermal control discussed but not implemented, too much power. Optical systems have to deal with temperature variations. Optical systems struggle to meet performance.
    • Systems free up resources and a clever solution of a “thermal oven” for critical optics is implemented
Ops Driven Thermal Requirements (2)

- **Example:**
  - **JWST**
    - Orbit at L-2 driven by high sensitivity of instruments
    - Detector technology necessitates cryogenic temperatures
    - Huge deployable sunshield
  - **Solar Probe Plus**
    - First mission to fly on Sun’s corona. Feeling hot, hot, hot!
    - The entire architecture is driven by the harsh environment
Working with Systems

• Above all else ask questions!
  – Communication is key, maintain good dialog
  – Don’t just expect to be handed requirements. Get involved in developing them.
  – It is important to understand what really matters, don’t get hung up on secondary minor issues, stay focused
  – Requirements are not sacrosanct, negotiate. The objective is for the system to work, not just your subsystem.

• Example: LRO lunar eclipses
  – Lot of back-and-forth about what lunar eclipses we had to design to survive
  – Systems originally wanted to design for any and all eclipses, high complexity
  – Negotiated to survival of only the predicted eclipses in the baseline (1 year) and extended (1 extra year) missions
Consider All Lifecycle Phases

Include all phases of lifecycle.
• Be mindful electrical and thermal dissipation are not always the same
  – Many s/s do not thermally dissipate all the electrical power
  – Example: Power Distribution Unit
    • Takes in 28V distributes secondary voltages
    • Ideally all power in goes out, i.e., 100% efficiency
    • Clearly efficiency is not ideal so only the conversion losses are dissipated in the box, in addition to internal dissipation from its C&DH

• Conservative power estimates
  – Not uncommon for component PDLs to worst-case power dissipation during analysis
  – Leads to lots of radiator to handle high loads …
  – … which then gets too cold in eclipse requiring lots of make up heaters…
  – … and then needs more power which is not cheap.
  – Again, good dialog and communication of assumptions is necessary!
Electrical vs Thermal Dissipation (2)

- Example of power history over development
  - Not uncommon for system to come in under estimates given “stacked” worse-case assumptions
  - Conservative, yes, but there’s a price to be paid

![Aquarius Radiometer Subsystem Power Technical Performance Measure](chart.png)
STOP Analysis

- **Structural Thermal Optical (STOP) Analysis**
  - Critical function for instruments
  - By necessity a highly integrated systems activity
  - Good coordination and documentation a must

- **Example: JWST**
  - A preliminary STOP analysis was needed to derive requirements for the primary mirror adjustment stages
  - TVac testing later validated that analysis
Verification vs Validation

• **System Verification**
  – Did I build the thing right?
  – Example:
    • After H/W build
      – Pass/fail tests against requirements

• **System Validation**
  – Did I build the right thing?
  – Should occur throughout all phases of development
  – Example:
    • During Requirement Development
      – Do I have the right requirements?
      – Are the requirements complete and correct?
    • On-Orbit, e.g., commissioning phase
      – Are we getting the science we needed?
NASA Mission Class

• NASA Mission Risk Classifications are established by NPR 8705.4
  – Considers factors like criticality to the Agency Strategic Plan, national significance, availability of alternative research opportunities, success criteria, magnitude of investment, and other relevant factors.

• Class D: Low Priority, High Risk
  – Rapid development, proof-of-concepts, lower costs
    • E.g., ISS payloads

• Renewed emphasis on risk-based decisions
  – Many of our “standard” rules or processes are consistent with Low Risk
  – IF agreed by Agency, can/should adjust for a higher risk posture
    • Have to be addressed during Formulation Phase NOT after
Class D Examples

• NASA Goddard developed the Class D Constitution
  – The application of processes and policies adapted for a design- and build-to-requirements approach and the risk aversion associated with higher classification missions (Class C, B and A) is having a negative impact on cost and performance for Class D missions.
  – The underpinnings of the Class D initiative are:
    • Greater attention upfront to the credibility of proposals
    • Clear and focused lines of accountability
    • Short reporting and communication channels
    • Ownership by the team
    • Expert advice and stewardship

• Examples:
  – CATS: Build-to-cost approach
    • Leverage existing instrument designs
    • Use commercial parts where possible
  – NICER: Operational flexibility
    • Point-anywhere observations but limiting when due to thermal
KEY ISSUES IN SPACE SYSTEMS
Key Issues– Heritage vs New Developments

• Using the word “heritage” is probably one of the quickest ways of getting in trouble and starting off the wrong foot when it comes to resource planning.

• “In the space business, no two systems can be the same; heritage is imperfect at best and usually less than we think.”
  
  – *From “Some Things You Always Needed to Know About Systems Engineering but Didn’t Know You Needed to Know”, M.G. Ryschkewitsch, Schneebaum Colloquium GSFC, September 2008*

• It is more prudent to use prior designs as a starting point for the Requirements and Basis of Estimate. This is very different than bypassing our homework on the basis that “it’s been done before”.

• EVERY claim to heritage must be substantiated and presumed not to be until shown otherwise.
Key Issues – “Good Enough”

- The objective is to achieve convergence according to the evaluation criteria.
  - “Make it work” = technical and performance
  - “Make it safe and reliable” = risk
  - “Make it affordable” = cost and schedule.

Overachieving is not necessarily a good thing. As PDL, you need to find the design sweet spot.

Optimization is a SYSTEM issue, not per subsystem.
Good Enough Example

• Example:
  – TIRS radiator
    • Cryocooler-based system with many unknowns at the time
    • Conservative estimate of cryocooler performance and thermal parasitics led to a rather large radiator with half or more of it blanketed for launch. Not optimized but the right decision for a schedule-constrained project.
  • How about TIRS2?
    – It is a lot more cost effective to do an exact repeat of the radiator rather than “optimize” it given its large blanketed area. Again, not optimized but the right decision for a cost-constrained project.
  – Use of off-the-shelf hardware
    • LRO over-sized reaction wheels
      – Developed by GPM
      – Overkill in performance and not “optimized”
      – Availability made it best choice for a schedule-constrained project
Over or Under Constraining the Solution Space

Under-constrained Solution Space
- Usually results in too many options being considered.
- Target may be hit or missed.
- Potential for large gyrations on the design.
- Cost and schedule inefficient.

Over-constrained Solution Space
Perhaps minimal gyrations on the design, which appears cost and schedule efficient, but may never have a chance to hit target.
“Optimally” Constraining the Solution Space

Optimally-constrained Solution Space
Reduces gyrations on the design over time, allowing control of cost and schedule while assuring we can hit the target.
• Primary measures: mass, power
  – Mass is cost!!!
  – Power generation requires mass, power system components are generally long-lead items.

• Other significant measures: data rate, pointing
  – Data rate impacts memory, processor, downlink (number of ground station contacts, downlink rate, transmitter power)
  – Pointing impacts type of sensors and actuators, which impacts mass and power

• Allocate and track
Key Issues - Safehold

- Out of power and it is dead.
- If it cooks or freezes, it will be dead.
- If we can’t talk to it, it might as well be dead.

- So, safehold must:
  - Be power and thermally safe
  - Allow communication
  - Be more reliable/robust than normal mode: it is your safety net
Key Issues - Margin

• Covers the unknown and the unknowable.
• Launches are substantially riskier than flying in an airplane, because rockets have very little margin:
  – An unexpected event or failure usually means the end of a space mission, while an aircraft can usually get back to the ground safely.
  – Space missions cannot afford the margin of a commercial airliner—too much energy is required to get to orbit.
• Margin costs money, so there is always pressure to reduce margin.
• Use good judgment and the experience of others to judge how much margin is appropriate.
  – Margins must be reasonable and commensurate with mission risk posture.
FINAL THOUGHTS
Lessons Learned

• Most problems are due to poor communication.
  – Create mechanisms for raising issues/concerns promptly and bring them to closure.
• Interface tests early and often will save you money.
• Don’t count on people reading documents.
• It’s the things you aren’t looking at that bite you in the rear.
• Discrepancies and disputes are an indicator of a mistake or a misunderstanding. Don’t brush them off—probe into the matter.
• You will re-learn some lessons others have already learned—try to keep that to a minimum as best you can.
Some More Advice

• Always look for the things you haven’t thought of, especially during testing.
• Perform tests to make sure it will work, not just because someone told you that the test is required.
• Test it in the configuration in which you will fly it.

• Always ask yourself if your answer makes sense.

• Always ask questions—understand why. No matter what you do, you can make better decisions if you understand the situation better.