Introduction to Systems Engineering for PHM: Module 1

Overview and Requirements Management

Ravi Rajamani

2018
Food for Thought

“He who can, does. He who cannot, teaches.”
– George Bernard Shaw (Man and Superman)

Somebody modified it to:
He who can, does. He who cannot, manages.

We can modify it further:
He who can, does. He who cannot, manages. But, with Systems Engineering, he makes fewer mistakes!
Ravi Rajamani, PhD, FSAE, FIMechE

Dr. Ravi Rajamani is an independent consultant who has accumulated years of experience in the area of aerospace propulsion and energy, data analytics and model-based methods for controls, and PHM.

Author
• Electric Flight Technology: The Unfolding of a New Future
• Two more books and 8 book chapters
• Journal and conference papers, and 26 patents.

Education
• BTech from IITD; MS from IISc; MBA from Uconn; PhD from UMN

Experience
• Meggitt PLC; United Technologies Corporation; General Electric Company

Professional activities and honors
• SAE technical committees and PHM Society,
• Visiting Professor, Cranfield University.
• Editor-in-chief of the SAE International Aerospace Journal.
• Fellow SAE International and Fellow Institute of Mechanical Engineers, UK.
• Forest R. McFarland Award, 2018
Prognostics & Health Management (PHM)
Overview

• An end-to-end capability that uses sensors, electronics, & analytics both on-board and off-board to accomplish:
  — Diagnostics: Determining the current health condition
  — Prognostics: Predicting the future state
  — Health management: Managing the asset based on this (and related) information

• PHM systems comprise of:
  — Sense, Acquire, Transfer, Analyze, Act/Display

• PHM systems manage assets while
  — Delivering guaranteed performance
  — Increasing availability
  — Lowering life-cycle costs
  — Without compromising system safety
Complex System With Many Moving Parts
Complex System With Many Moving Parts

ONBOARD

TRANSPORTATIONAL

OFF BOARD

ENTERPRISE LEVEL
Problem with Current PHM Systems

• Developed “after-the-fact” to correct problems
• No systematic development effort
• Budget subject to whims of the larger program
• Return on investment might take a long time
• Not many skilled personnel with experience
• Many subsystems not “health-ready”
Part of the Solution

• Have a dedicated manager/team for PHM
• Co-develop with the rest of the system
• Protect budget as much as possible
• Develop shorter-term “off-ramps” if possible
• Use more modern systems engineering practices
• Recruit for the job; not poach from other departments
• Follow standards and insist on “health-readiness”
• ...
• ...
A Step in the Right Direction: Systems Engineering
What is a System?
Definition of a System

• ...a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce systems-level results. The results include system level qualities, properties, characteristics, functions, behavior and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected (Rechtin, 2000).

incose.org (consensus of the INCOSE Fellows)
So, a System is:

- A complex interacting set of resources
  - H/W, S/W, facilities, processes, etc.
- Hierarchical (generally)
  - Consisting of subsystems
  - Related components
- Serving a purpose
  - i.e., responding to some expressed need
  - Value added by the system is more than that contributed by the individual parts
A System Can be This
...or Even This!
Types of Systems

• Natural and Man-made
• Physical and Conceptual
• Static and Dynamic
• Closed and Open

• EXAMPLES?

Broad Characteristics

Broad Characteristics

SYSTEM

Technological
Economic
Social
Political

Articulation of customer needs

Human
Equipment
Software
Materials

Satisfaction of customer requirements

System of Systems

Diagnostics and prognostics data support system for an airline

- Flight Data System
- ACARS System
- SatCom System
- Ground Operations
Systems Engineering
Current Environment

Discipline of Systems Engineering (SE)

• SE is an interdisciplinary approach and means to enable the realization of successful systems. (INCOSE, 2004)

• SE is an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near optimal manner, the full range of requirements for the system. (Eisner, 2008)

• SE is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variable and relating the social to the technical aspect. (FAA 2006)

History of Systems Engineering

• Early 1940s: Systems engineering as a discipline started in Bell Labs. Within Bell Labs its practices date back even further.
• 1946: US DoD creates the RAND Corporation. They worked on developing Systems Analysis, a part of SE.
• Late 1940s: DoD uses SE for the development of missiles and missile-defense systems.
• 1950: Gilman, from Bell Labs, taught the first SE course at MIT.
• 1990: NCOSE (National Council on SE) was established, which later became INCOSE in 1995. [www.incose.org](http://www.incose.org)
• 2013: INCOSE launched Wiki called Systems Engineering Body of Knowledge (SEBoK). IEEE and SE Research Center (SERC) help manage this today. [www.sebokwiki.org](http://www.sebokwiki.org)
Why use Systems Engineering?
Wally, we don't have time to gather the product requirements ahead of time.

I want you to start designing the product anyway. Otherwise it will look like we aren't accomplishing anything.

Of all my projects, I like the doomed ones best.
Cost of fixing errors during different product stages

Reference:


“Error Cost Escalation Through the Project Life Cycle,”
4th Annual International Symposium of INCOSE; 19-24 Jun. 2004; Toulouse; France

Upper – Upper limit of the range
Lower – Lower limit of the range

Data consolidated over many NASA projects. Similar results for other fields as well.
Example

- EHM System
  - Designed without taking the long-term view
  - No thought given to use of system in a safety-critical environment
- Key problem: Did not consider all stakeholders
- Regulatory authorities will not allow system to be used outside of design parameters
- Example of engine with fan blade retention problem; FAA would not allow EHM system to be used. Insisted on regular – and frequent – inspections, thereby resulting in higher costs.
Program Lifecycle
Vee-Diagram

- Concept stage
- Requirements development
- Preliminary design
- Detailed design
- Prototype
- Validation
- Verification
- Integration
- Qualification test
- System integration
- Production
- Design assurance test
- System verification
- Design assurance test
- System integration
- Production
Program Milestones

Typical reviews

• Concept review
• Requirements review
• Preliminary design review
• Critical design review
• Test readiness review
• Qualification test review
• Production readiness review
• Final acceptance review
Requirements Development
Writing Good Requirements

A well written requirement should be simply written, unambiguous, and easy to interpret.

Different people can write it differently. But everybody should be able to interpret it the same!
Standard Construct

• Follow this simple construct:

  • **Who shall [verb phrase] what [under what constraints]**

    • Who = Subject of the requirement (i.e., the system)
    • shall = Specifies that the requirement is essential.
      • If the requirement is not essential, use “should”
      • If the requirement is a statement of fact, use “will”
    • what = The object of the verb phrase
    • [constraints] = Used only if the requirement is conditional

• Example:
  • The ODM shall store debris count numbers for the past 100 flights.
## Characteristics

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Necessary</td>
<td>Only include if the system cannot meet real needs without it.</td>
</tr>
<tr>
<td>Verifiable</td>
<td>Every requirement should be verifiable, and the verification method and level at which the requirement can be verified should be documented.</td>
</tr>
<tr>
<td>Unambiguous</td>
<td>Make sure there is only one interpretation for the requirement.</td>
</tr>
<tr>
<td>Complete</td>
<td>All known requirements should be stated. Also, all conditions under which the requirement applies should be stated.</td>
</tr>
<tr>
<td>Consistent</td>
<td>A requirement should not conflict with another.</td>
</tr>
<tr>
<td>Correct / Traceable</td>
<td>Verify that the requirement truly captures a need. Ensure that its linked to a customer requirement or a need.</td>
</tr>
<tr>
<td>Concise and Singular</td>
<td>The statement should state only one requirement, it should be stated simply and clearly, and should not be global. Make sure it is in active voice.</td>
</tr>
<tr>
<td>Standard constructs</td>
<td>Are the requirements stated as imperative needs using “shall,” optional needs using “should,” and statements of fact or declaration of purpose using “will.”</td>
</tr>
<tr>
<td>Feasible</td>
<td>Make sure requirement is achievable given the imposed constraints of cost, schedule, technology, and risk.</td>
</tr>
<tr>
<td>Implementation Independent</td>
<td>Ensure that the requirement describes functions, characteristics or constraints without describing a solution.</td>
</tr>
</tbody>
</table>
Examples
Example 1

• The AC input shall be electrically isolated from the DC output (including the DC output return).
Example 1

• The AC input shall be electrically isolated from the DC output (including the DC output return).

• The system shall electrically isolate the AC input from the DC output.
• The system shall electrically isolate the AC input from the DC output return.
Example 2

• The system shall be prohibited from half-wave rectifying AC power.
Example 2

• The system shall be prohibited from half-wave rectifying AC power.

• The system shall perform full-wave rectification of the AC power input.
Requirements Flowdown
Requirements and Architecture are Interlinked
# A Logical Process

<table>
<thead>
<tr>
<th>Process</th>
<th>Inputs/Outputs</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Requirements Review</td>
<td>Review the source requirements and user community needs to resolve conflicts and identify constraints. This may include setting performance metrics.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Requirements Analysis</td>
<td>Develop use cases, characterise health parameters, review information exchange and evaluate constraints.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Define Logical Architecture Options</td>
<td>Define operational scenarios, create logical architectures, build functional models and evaluate coupling between functions.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Constraints Analysis and Physical Architecture Options</td>
<td>Evaluate performance within known constraints, outline physical architecture options that could satisfy the functions and create a configuration for the system.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Functional Requirements Allocation to Subsystems</td>
<td>Map the logical to the physical architecture, identify context and formalise system functionality, develop a set of (preliminary) complete architectural options.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Architectural Design Evaluation/Validation</td>
<td>Develop and apply evaluation criteria (and associated weightings), rank solutions and down-select.</td>
</tr>
<tr>
<td>Step 7</td>
<td>Document</td>
<td>Document the chosen architecture, trade-study results and the lower level sub-system requirements (functional and non-functional).</td>
</tr>
</tbody>
</table>

ARP6290: Guidelines for the Development of Architectures for Integrated Vehicle Health Management Systems
## A Logical Process

<table>
<thead>
<tr>
<th>Process</th>
<th>Inputs/Outputs</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Requirements Review</td>
<td>Review the source requirements and user community needs to resolve conflicts and identify constraints. This may include setting performance metrics.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Requirements Analysis</td>
<td>Develop use cases, characterise health parameters, review information exchange and evaluate constraints.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Define Logical Architecture Options</td>
<td>Define operational scenarios, create logical architectures, build functional models and evaluate coupling between functions.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Constraints Analysis and Physical Architecture Options</td>
<td>Evaluate performance within known constraints, outline physical architecture options that could satisfy the functions and create a configuration for the system.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Functional Requirements Allocation to Subsystems</td>
<td>Map the logical to the physical architecture, identify context and formalise system functionality, develop a set of (preliminary) complete architectural options.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Architectural Design Evaluation/Validation</td>
<td>Develop and apply evaluation criteria (and associated weightings), rank solutions and down-select.</td>
</tr>
<tr>
<td>Step 7</td>
<td>Document</td>
<td>Document the chosen architecture, trade-study results and the lower level sub-system requirements (functional and non-functional).</td>
</tr>
</tbody>
</table>

ARP6290: Guidelines for the Development of Architectures for Integrated Vehicle Health Management Systems
Summary

• For the right sized project, systems engineering is critical. Takes about 15% of the budget, but can save orders of magnitude more.
• Most aerospace companies will not undertake a major program without systems engineering at the core of the program. Even more true of the DoD!
• Because PHM is a multidisciplinary undertaking, it is particularly important to use systems engineering in its development.
• Benefits include lower maintenance costs, higher supply chain efficiencies, and possibly increased safety (in the future), but
• ...regulations do not support implementation to take full advantage of PHM.
• A critical first step in the SE process is to get a good set of requirements.
• Writing good requirements is not difficult, but needs discipline.
• Having good requirements helps with developing the right architecture