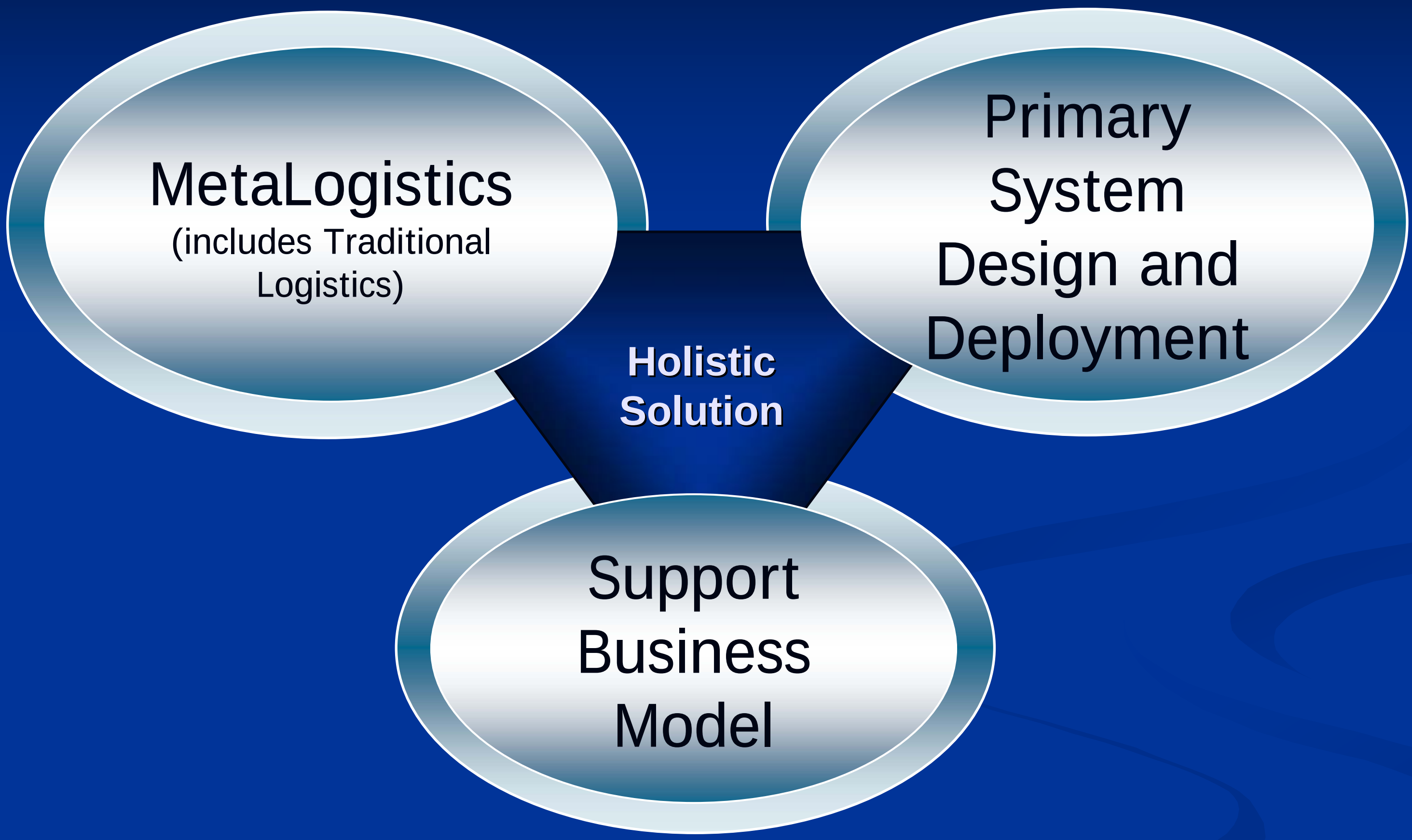
The Seven Affordability Sins of Logistics System Integration

Tom Herald, Ph.D.
Lockheed Martin Fellow
Simulation, Training & Support

Joe Bobinis, PMP
Lockheed Martin Fellow
Info Systems &Global Sustainment

Attributes of System Operational Effectiveness



The focus and funding are often centered on delivering tactical systems; however, a more holistic focus is on delivering mission capabilities.

How do we move our focus to “Mission” vs. “System”?

- As the DoD’s business model continues to evolve, its focus on meeting varying mission needs within a bounded O&S budget is pushing for some kind of an evolutionary business approach.
- Our Processes and Tools are “System (or Platform)-centric”
- What else is needed in order to perform a mission? What is driving increasing O&S costs, reducing much need modernization funds?
- Unfortunately, Enabling Systems are still often implemented AFTER the delivered tactical system and as an externally designed system. This approach has been successful for many years, but TODAY does not lead to the most AFFORDABLE and mission ready systems.

“Always design a thing by considering it in its next larger context – a chair in a room, a room in a house, a house in an environment, and an environment in a city plan.”

- Eliel Saarinen, Finnish Architect

Helsinki, Finland Railway Station

The Seven Affordability Sins . . .

- Sin 1: Insufficient customer needs analysis (i.e. dig deeper and broader)
- Sin 2: Belief that all requirements can be deduced before the system is deployed.
- Sin 3: Ignore the system requirements necessary to permit enabling systems success.
- Sin 4: Usability design that is engineering-centric versus user-centric.
- Sin 5: Designs without the human-in-the-loop considerations.
- Sin 6: Acquisition cost focused.
- Sin 7: Limited consideration for net-centric environment integration.

Approaches for improving the affordability of mission success, through a more holistic approach for designing complex systems

Sin 1: Insufficient customer needs analysis (i.e. dig deeper and broader).

- **Issue:** By not digging to the 'root need', an incorrect support enabling system solution may result.
- Learning opportunities:
  - **'Carrier Electro-Magnetic Radiation Signature'**
    - Accurate Design Algorithms
    - Successful Design implemented on a helicopter platform
    - BUT: Puts humans in harms way (Helicopter Pilots say . . .)
    - Technical merit beauty, but failure to meet holistic operational expectations.
  - **Aerial Common Sensor ISR Army Aircraft**
    - Requirements creep to include cross-service utilization with Navy
    - More customers often means more needs
    - Tends to cumulatively add to functionality versus integrate easily with existing functionality (a bolt-on functional mentality)
    - This program was terminated due to unacceptable growth in weight, that drove increase in cooling and power requirements that became a negative viscous cycle.

Sin 1 Conclusion . . .

- Requirements Management (creep) is often cited as a root cause for unsuccessful programs.
- Wait a minute! We're smart engineers, we know that requirements creep is a problem, So WHY does it keep happening?!
  - No schedule time for sufficient customer needs analysis
  - No holistic enabling system (support) integration
- Result: The Primary and Support Systems are not integrated and thus the requirements evolve separately.
- Thought: What about the "Development Environment System?"

Discovery and evolution through the design phase is natural. Needs analysis done well accelerates functional discovery AND minimizes unforeseen requirements creep.

Sin 2: Belief that all requirements can be deduced before the system is deployed.

- **As an extension of Sin 1:** Even excellent needs analysis may still assumes a-priori knowledge of the full breadth of the operational uses, environments, laws, etc. This is typically NOT a reasonable assumption.
- Learning opportunities:
  - **Acoustic Rapid COTS Insertion (ARCI) Navy program**
    - Requirements are the variable; cost and schedule are locked
    - System obsolescence (support) and functional growth are merged
    - The system to evolves capabilities annually
  - **Aircraft weight grows at the rate of '1-pound per day of deployed operations'**
    - Bolt-on functionality growth approach
    - ~ 300 pounds per year for 20 years = 6000 pounds!
    - Also additions to size, weight, power, cooling, logistics footprint.
    - Knowing this military history, do we design for this in mind??

Sin 2 Conclusion . . .

- Clearly define a tradable space for system evolution that becomes your decision algorithm for changes.
  - Consider Life Cycle Cost, Reliability, Risk, and Performance as a 4-dimensional trade space as a means of managing growth requests.
- Ensure that the architecture is truly open and permits evolution of the underlying hardware and software physical solutions

Army OODA Loop – In battle situations there is a constant loop of Observe, Orient, Decide and Act. Continuously manage emergent information.

Suggestion: Instead of fearing requirements creep, we should embrace the dynamic nature of a system design through incremental, spiral, and agile development methods.

Sin 3: Ignore the system requirements necessary to permit enabling systems success.

- **Issue:** Enabling systems often do not get ‘equal design focus’ and yet the impacts of a flaw in the enabling systems are often program show-stoppers
- Learning Opportunities:
  - F-117 Nighthawk – First stealth fighter
    - Disruptive technology that revolutionized battle options
    - Still one of the finest, most technologically-advanced fighters in aviation history.
    - World-class mission capabilities as evidenced during Desert Storm
    - The initial design focus was stealth fighting capability, quickly.
    - The enabling system operational consideration was given a “back-seat”.
    - The enabling systems also have world-class records with keeping the aircraft flying; however, the costs for this support are quite high. .
  - New environments (Desert to Rain Forest to South Pole), New uses (unforeseen requirements), and Emerging threats
    - These conditions can take an apparently successful system solution and render it unsuccessful. Desert Storm was an eye-opener for the assumption that performance and reliability were the same in a high-grit, high-heat environment.
    - Getting the enabling system materiel “in country” was efficient, but made useless because the enabling system was not designed to get them to “point of use”

Conclusion 3 . . .

- Does your system requirements management database have derived supportability requirements included?
- Typically not; however, the high-level supportability requirements are often delineated in the Originating Requirements Document or the Statement of Work.

The "best performing system design ever" will still fail if the consumables and logistics tail are not sufficient to ensure system Operational Effectiveness.

Integrated design for support – Supportable design – Support the design affordably

Sin 4: Usability design that is engineering-centric versus user-centric.

- **Issue:** Designers are too often enamored with functional elegance and flexibility making everything in the solution a variable; however this demands too much user interaction and intimate process knowledge in order to properly provide inputs and interpret system outputs.
- Learning Opportunities:
  - **MOP4 operations**, where soldiers are wearing Chem-Bio suits
    - Allow for system operations with bulky gloves
    - Extreme environments, fatigue, heat, cold, etc.
  - Move toward **Autonomic Logistics** versus traditional support options
    - More system integration
    - User-centric designs and focus, versus functional decompositions
    - Learning systems versus static systems

Conclusion 4 . . .

- Marine Corp Embedded Platform Logistics System (EPLS) Gene Morin, Program manager is quoted as saying, **“I want my Marines to have their fingers on triggers and not on keyboards.”**
- This says it concisely. Logistics support systems should “make things happen, when they need to happen, and without human intervention if at all possible”.

Suggestion: Carefully trade functional flexibility with user simplicity. To this end, possibly consider multiple modes, Users and Use Cases. WHEN at a minimum? At all Design Reviews.

User-centric design methodologies, and using the recommendation from Sin 1, ensure that deep dive analysis distills the relevant information which can be absorbed in a “User glimpse”.

Sin 5: Designs without the human-in-the-loop considerations.

- **Issue:** Major Total Ownership Cost (TOC) and System effectiveness are driven by Humans in the System
- Learning Opportunities:
  - **Air Bag Design (initial designs)**
    - Requirements defined in early ’80’s and deployed in early ’90’s
    - Design for men only (50 Percentile Male)
    - The Air Bags themselves were causing female fatalities
  - **U2 Spy Plane***
    - Disruptive technology with landmark capabilities
    - 70,000+ ft altitude and extended loiter times*
    - Requires space suits for pilots, no relief, no physical movement possible, etc. The pilot was a back-seat consideration.

Would we design the U2 today or might we use a UAV for the mission?

* Source: www.wikipedia.com

Conclusion 5 . . .

- Fight the tradition of 'the way things have always been done', and intentionally put the human inside your system design requirements boundary.
- It may then become obvious that the human is being expected to do 'too much', and therefore, the design team should explore automated and autonomous alternatives for the system solution.

How does the human interact with your system?
Human at risk? Human overloaded (information, attention, actions)?

You may also discover that the maintenance and upgrade for Automation is much cheaper than the humans that are freed up.

Sin 6: Acquisition cost focused.

- **Issue:** A focus on acquisition cost alone when making design decisions is a typical approach; however, leaves much affordability opportunities unleveraged.
  - 70% of the O&S costs are determined as soon as requirements are set.
  - Wait a minute . . . Isn't this a good thing?

- Learning Opportunities:
  - **F-35 Multi-national and Joint-forces Fighter:**
    - Mission Reliability (Operational Availability) is a Key Performance Parameter (TOC too)
    - KPP's for: Sortie Generation Rate and Logistics Footprint
    - $135 B or a 56% estimated TOC savings compared to legacy systems
    - Mission Reliability of over 90% and a 30 day self sustained mission
    - 12% or $16B is expected to come from Enabling System Automation Prognostics & Autonomics
  - **Advanced Amphibious Assault Vehicle:**
    - Acquisition program focused on mature technology and O&S costs.
    - Program office co-located with contractor facility and extensive use of end user assessment of system operational effectiveness.
    - Extensive reliability testing with common components of other Marine Corp. weapons systems.
    - Initial O&S cost savings = $29 million.

Conclusion 6 . . .

- Evolutionary Acquisition approach to design.
  - Incremental development which assumes Life Cycle as a requirement of the Enabling System AND also as a Mission Requirement of the Primary System.

- The design “end state” needs to include the system life cycle through to disposal.

TOC and Performance must be of equal importance in the design trade space.

Move is from a ‘point solution’ (Performance) to an ‘evolutionary solution’ perspective.

Sin 7: Limited consideration for net-centric environment integration.

- **Issue:** In a typical development environment, the design team focuses on the requirements for which they are paid to innovatively solve. Also typically, this is viewing the system as a stand-alone entity with interfaces to the world around it.
- The challenge is not as simple as ‘does this system talk to that system’ but rather the emergent system of systems capabilities and challenges that occur when systems are connected within a network-centric environment.

Conclusion 7 . . .

$$P_{NCS} = \sum_{i=1}^{N} P_i + \sum_{\substack{i=1 \\ j=1}}^{N} \left(\sum_{k=1}^{M} (P_i \cap P_j)_k \right);$$

For all $i \neq j$ AND where $k > 0$ (i.e. the system pairing has connectivity)

- P_{NCS} ■ The total performance of the Network-Centric System
- P_i ■ The performance capability of a Stand Alone System (no network connection)
- P_j ■ The performance capability of a Stand Alone System (no network connection)
- N ■ The number of Independent Systems (Network Nodes)
- M ■ The number of Independent functional connection paths for a P_i and P_j pairing
- $P_i \cap P_j$ ■ This intersection represents the resultant performance from the system connectivity, which could be Zero if there is no system advantage or detractor, Positive if the connectivity advantages the ConOps (Mission Needs) or Negative if the connectivity is not required by the ConOps (i.e. outside of the mission performance boundaries)

The net-centric whole is greater than the sum of the systems.
Pairwise additive capabilities, Triples additive capabilities, . . .
Some capabilities are Good and some are Negative in YOUR system.

Principal Recommendations

- Focus on the mission needs, make time for and dig deep to root out the true stakeholder needs.
- Ensure the system is affordably evolvable through the support life cycle.
- The Primary and Enabling Systems must be holistically designed as a single complex System of Systems.
- Document and decompose ALL of the requirements.
- A combined and equal focus of performance and support requirements during design for total system performance responsibility.

Thought: These recommendations outline a more inductive approach to our traditionally deductive engineering paradigm.