Applying Architectures to Modernization, Sustainment and Investment Planning

Rick Sorensen

Chief Systems Engineer

KIHOMAC, Inc.

Abstract

In-service legacy systems are being tasked to meet emerging demands for which they were not originally designed. At the same time, sustaining the existing system capabilities, in a fiscally-constrained environment, still requires some financial investment as well as updated planning to address the systems' changing environments.

Meeting the combined needs for both modernization and sustainment presents several difficult challenges for the systems engineer. The question is, how do we sustain existing capabilities while accommodating growing demands on existing infrastructure? We need processes, tools, and methodologies in place to iteratively and repeatedly assess systems engineering budgeted areas such as space, weight, power, datalink bandwidth, data bus bandwidth, processor and memory size/growth, in light of multiple modification actions. This on-going assessment is a critical aspect of the systems integration efforts needed to recognize and mitigate future limitations and to provide the best insights into the types of investments required.

When we combine systems engineering practices, an "as-is" architecture, and a "to-be" architecture, we can begin to forecast impacts on the many systems engineering budget areas. This enables us to anticipate and plan for mitigations to impacted areas so that continued modernization and sustainment is feasible. The holistic view offered by implementing a mature architecture process provides cost-effective solutions that facilitate long range planning. At the same time standardized processes enable the continued sustainment and modernization of legacy systems to meet the needs of the users.

This paper presents an overview of the processes and methodologies we currently apply to the development of a "to-be" architecture to document sustainment and modernization requirements. It also addresses the support provided by these activities into the investment planning process that is necessary to execute ongoing sustainment and modernization planning.

Agenda

- Why Integrated Solutions?
- The Engineering Baseline
- Sustaining Existing Capabilities
- Addressing Emerging Needs
- Integrating Sustaining & Emerging Needs
- To-Be Architectures
- DoDAF Support for Planning
- Conclusion
- Questions & Answers

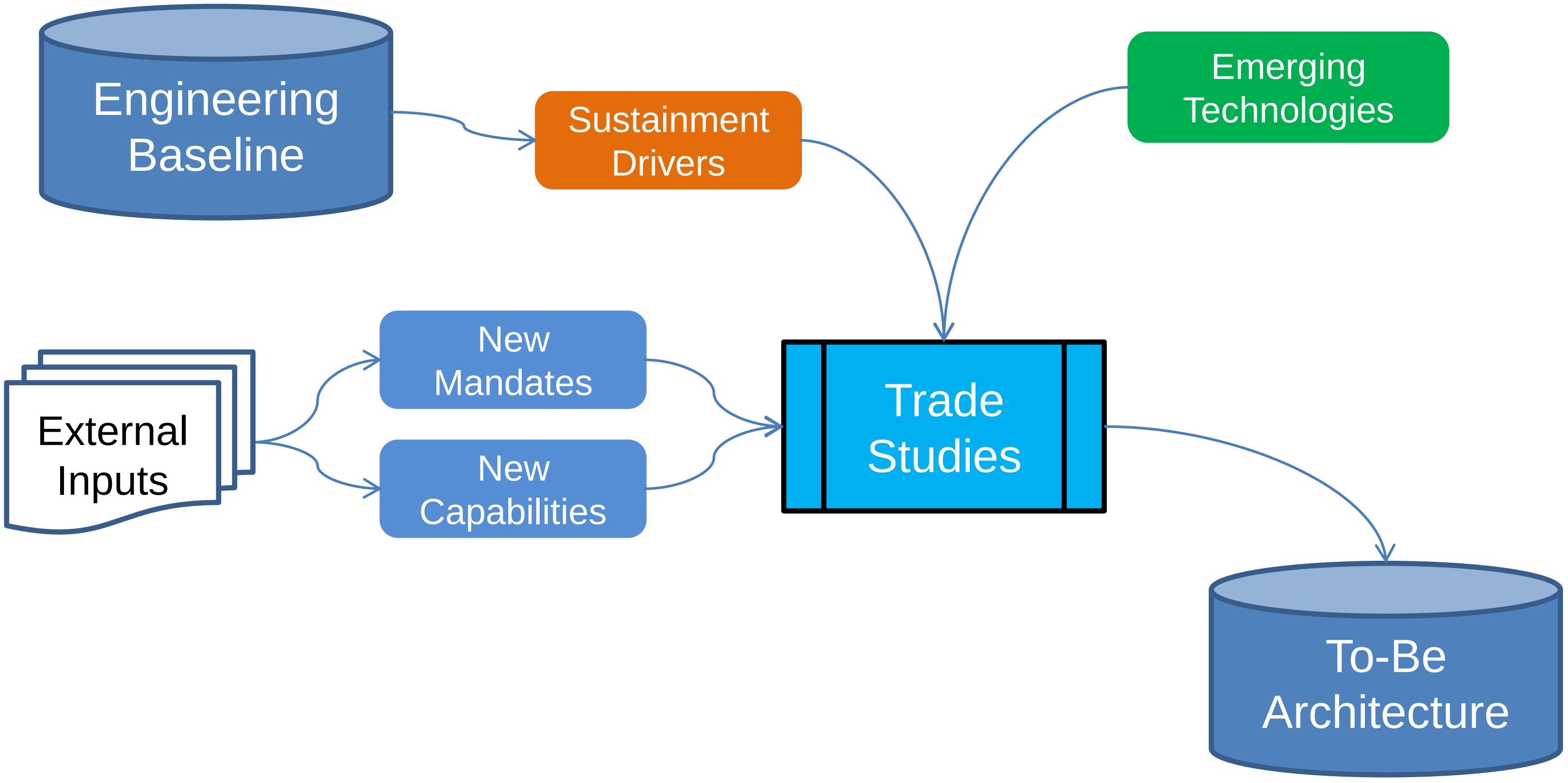
Integrated Solutions in Sustaining & Modernization Efforts

Why Integrated Solutions?

- Long Term Cost Benefit
  - Often higher initial cost over Form/Fit/Function (F^3) modifications
  - Payoff is in:
    - Support cost
    - Reliability growth
    - Improved A_o
    - Added capability

- Systems Engineering budgets
  - Maintaining SE budgets
  - Providing growth for future needs

The Process


Engineering Baseline
Sustainment Drivers
Emerging Technologies
External Inputs
New Mandates
New Capabilities
Trade Studies
To-Be Architecture

Capturing the Engineering Baseline

Legacy Systems

Engineering Baselines

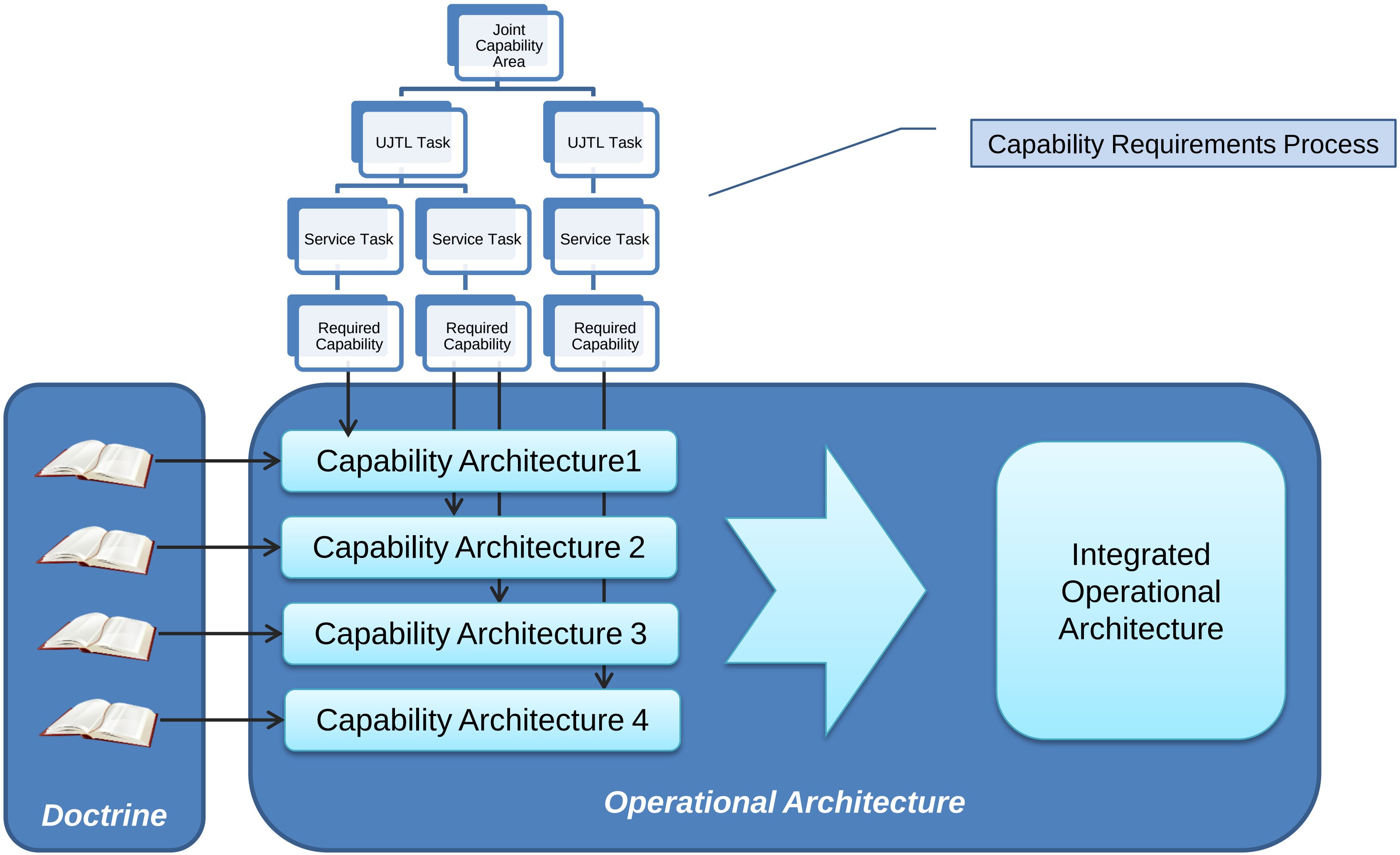
- Engineering Baseline is the point-of-departure
  – What is “IT” today?
  – How well does “IT” meet current required capabilities?
  – How are current capabilities implemented?

- Challenge with legacy systems:
  – How much of the engineering baseline exists in usable form?
  – How much exists as tribal knowledge?

Architectures

- Highly effective method for capturing an engineering baseline
  - Bring all the information together in one consistent, traceable form
- To-Be architectures (what this presentation focuses on)
  - Highly recommended!
  - Support for JCIDS processes
    - ICD, CDD, CPD
  - Growing requirements for deep insight and support from architectures
    - CJCSI 6212.01F (NR KPP) made fundamental changes!!

Operational Capabilities



New ‘take’ on traditional operational analysis leading to operational requirements

Requirements Buckets

Tying Operational and System Architectures together:

Operational Architecture

System Architecture

Why "Requirements"?

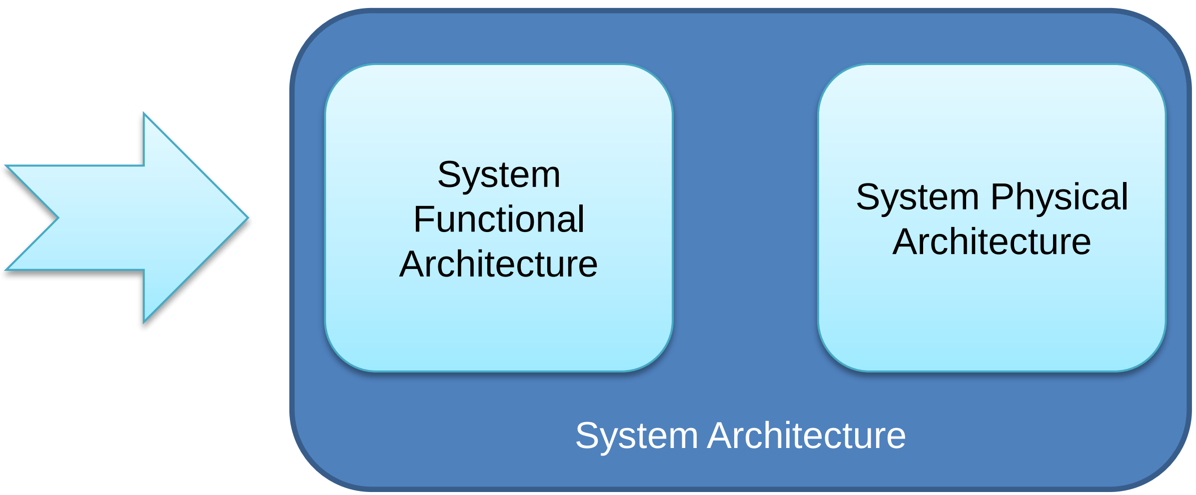
- I thought this was an architecture discussion. Why are we talking 'requirements'?

- Answer:
  - Ultimately, and especially with legacy systems, we have 'requirements' documents that are integral to contracting processes
  - Disconnecting 'architectures' and 'requirements'
    - Bad plan!

- However… there are no DoDAF 'Requirements Viewpoints'!! Huh…..

Realizing Requirements

Requirements



What Do We Get?

- Clear definition of why we have what we have
  - Why is this subsystem/LRU part of my design?
- What do we have to maintain? Why?
  - Its ALL about <u>maintaining capabilities</u>, not systems or subsystems or LRU’s
    - AND… we know exactly what capabilities
- Starting point for future planning!

Planning

Sustaining Existing Capabilities

- Existing capabilities
  - Usually well understood, sort of . . .
    - “Folklore” vs. authoritative documentation
  - We have a design (or designs) to achieve capability X . . .
    - But how well does the legacy design meet current needs?
      - How do we know?

Sustaining Engineering

- Sustaining existing capabilities
  - Identify current and near-term deficiencies
    - Responding to ‘brush fires’
    - Identify a solution
    - Execute acquisition and modification programs to implement
  - Well-understood process
    - Supported by current sustainment acquisition practices
  - Not always the best outcome!
    - Stovepipe solutions rarely achieve significant cost savings or reliability improvements

Emerging Needs

- Requirements in the ‘too hard’ box
  – When this system was built way back when, we didn’t know how to do xxxx!
- Requirements in the ‘too expensive’ box
  – When this system was built, we wanted to do xxxx, but it wasn’t in our budget
    - Or it was, initially, and we postponed it because it was going to be too expensive
    - Or it was and…..
- ‘Totally New Concept’!
  – Add new capabilities to existing system xxxx
    - Such as integrating legacy (standalone) systems into the GIG

Process

.. expanded

New Development

- Classic Systems Engineering Process:
  1. Operational Architecture (OA)
     - Result: Operational requirements documented with users context, desired behavior, KPP’s, KSA’s, and MOE’s
  2. System Functional Architecture (SFA)
     - Result: Operational requirements developed as necessary behavior
  3. System Physical Architecture (SPA)
     - Result: System behavior developed as a physical solution, traceable back to system, operational, and capability requirements

Addressing Sustainment

- Sustainment Sequence:
  - The physical item(s) to be modified/replaced are known, so:
    1. Identify the necessary functionality (reverse engineer in System Functional Architecture)
    2. Verify the operational need (trace functionality to operational requirements in Operational Architecture)
    3. Re-allocate functionality (may not be the same allocation as previous)*
       - In typical sustaining engineering, no re-allocation is done. New boxes implement previous functionality 1:1

* If you are only doing sustainment, proceed with this step. IF you have emerging requirements also, hold off on re-allocating until the full scope of required functionality is identified.

Addressing Emerging Needs

- Developmental Sequence:
  1. Identify new capabilities (Update Baseline Operational Architecture)
  2. Identify new functional requirements (Update Baseline System Functionality Architecture)
  3. Re-assess and allocate to recommended physical implementation (Update Baseline System Physical Architecture)*

- Result:
  – To-Be Architecture, based on current Baseline
    - Not the new Baseline until modifications are executed!
  – 1-n recommended modifications, with
    - Documented dependencies,
    - Technical detail to support cost assessment,
    - Recommended execution process, with architectural support (DoDAF views) for JCIDS processes

    - More details

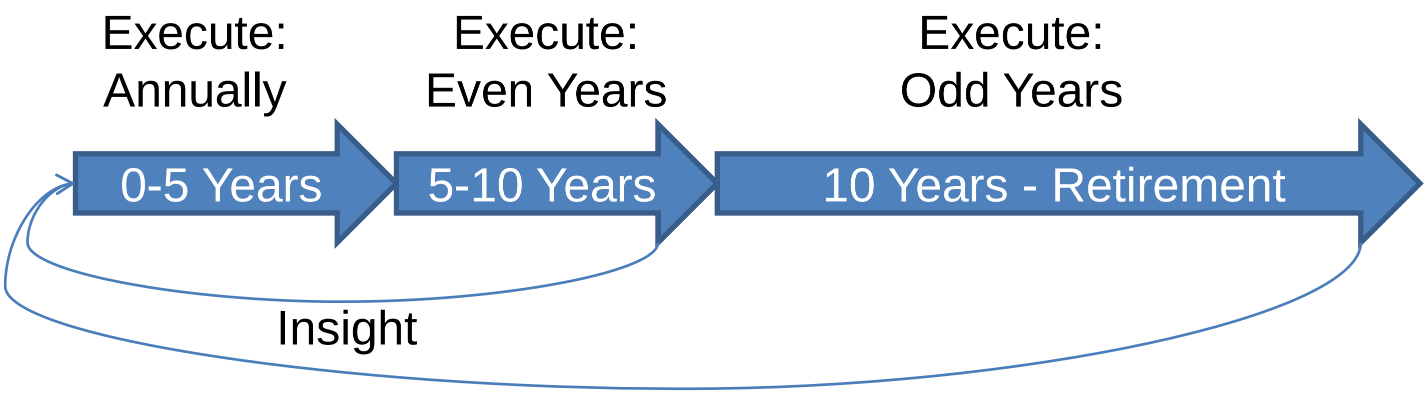
* If you have sustainment requirements to address, re-allocating sustainment functionality at this point..

To-Be Architecture Thoughts

- Forward looking / To-Be Architectures
  - Are not one-size-fits-all
  - Need to be purpose-driven
    - In this presentation, ‘purpose’ is to identify those system aspects that are expected to require investment and to provide systems engineering insight into the investment planning

- Recommendations:
  - Establish three To-Be architectures:
    - Near-term
    - Mid-term
    - Long-term (to retirement)

To-Be Architectures



Near-term To-Be Architecture

- Timeframe 0-5 years
  - Revisit annually to support POM process
    - Address implications of Mid-term architecture
    - Address systems engineering budget forecasts
      - Space, weight, power, CPU utilization, memory utilization, bus bandwidth, etc.
  - Focus is on establishing acquisition, execution strategy
  - Detail level should focus on which HW & SW items are we buying, building, and modification planning
  - Confidence level should be High

- Benefit:
  - Facilitates upcoming POM cycles
  - Facilitates modification planning
  - Facilitates OA & OUE investment planning

Mid-term To-Be Architecture

- Timeframe 5-10 years
  - Revisit every other year
  - Focus is on emerging technologies, retiring technologies, known policy & guidance changes
  - Detail level is on systems & subsystems
    - NOT at the HW/SW level
  - Confidence level should be Medium – High

- Benefit:
  - Facilitates Near-term planning
    - Impact on SE budget forecasts
    - Insight into emerging technologies

Long-term To-Be Architecture

- Timeframe 10+ years
  - – Revisit on off-years (alternate between #2 & #3)
  - – Focus is on emerging technologies, retiring technologies, known policy & guidance changes
  - – Detail level is on emerging architectural and design concepts
    - Example: ‘services’ vs. ‘systems’
  - – Confidence level should be Medium – High

- Benefit:
  - – Facilitates Near-term and Mid-term planning
    - Insight into emerging architectural concepts
  - – Facilitates retirement planning
    - Early insight into potential technical challenges

DoDAF Support

What DoDAF Views support To-Be architectures and investment planning?

Applicable Capability Views

- CV-1 Vision
  - Describes a Project's Visions, Goals, Objectives, Plans, Activities, Events, Conditions, Measures, Effects (Outcomes), and produced objects.

- CV-3 Capability Phasing
  - The planned achievement of capability at different points in time or during specific periods of time. The CV-3 shows the capability phasing in terms of the activities, conditions, desired effects, rules complied with, resource consumption and production, and measures, without regard to the performer and location solutions.

- CV-4 Capability Dependencies
  - The dependencies between planned capabilities and the definition of logical groupings of capabilities.

Applicable Project Views

- PV-2 Project Timelines
  - A timeline perspective on programs or projects, with the key milestones and interdependencies.

- PV-3 Project to Capability Mapping
  - A mapping of programs and projects to capabilities to show how the specific projects and program elements help to achieve a capability.

Conclusion

- To-Be architectures support
  - Near-term planning
  - Mid-term planning
  - Long-term planning

- Executing To-Be architectures in DoDAF
  - Support for JCIDS processes
  - Capability AND Project views
    - Common ‘dialog’ supports all stakeholders

Questions?

Thank You!

Contact

Richard L. Sorensen

Chief Systems Engineer

KIHOMAC, Inc.

334 N. Marshall Way, Suite J

Layton, UT

(801) 593-7088 ext 162

Richard.Sorensen@kihomac.com

Biography

Richard L. Sorensen is the Chief Systems Engineer at KIHOMAC Inc. He has over thirty two years' experience in systems engineering and systems architecture in both military and civil applications. His background includes hands-on system operations and maintenance as well as modeling and simulation, information systems architecture development and integration, business process reengineering, database design, hardware and software systems integration, and enterprise architecture planning.