



A Systems Engineering Approach to Architecture Development

*National Defense Industry Association
17th Annual Systems Engineering Conference
27-30 Oct 2014*

*David A. Di Pietro
Senior Systems Engineer
NASA Goddard Space Flight Center
Mission Systems Engineering Branch, Code 599
Greenbelt MD 20771
david.a.dipietro@nasa.gov, ddipietro@erols.com
301 286-0240*



D.A. Di Pietro
Goddard Space Flight Center

Published and used by NDIA with permission



Discussion Objectives

- **Using a simple extension of basic systems engineering (SE) practices**
 - 1) *Describe what a mission area architecture (MAA) is and show how it integrates into a Notional Civil Space (NCS) Architecture Framework*
 - 2) *Describe an effective approach for developing an MAA*
- **Note: The NCS Architecture is notional and is for illustration & context only – no such architecture has been defined**
 - *But, for this discussion imagine there is an NCS architecture*





Architecture Studies - Beginning Thoughts

- **Conducted prior to Pre-Phase A of project life cycle**
 - *Scope broader & shallower than scope for concept design studies in Pre-Phase A*
- **Can be conducted at mission area or mission level**
 - **MAA Studies:**
 - ❑ *Address best-value mix (number, capability, rough cost, etc.) of collection of MAA assets that work together in specific scenarios & time frames to accomplish mission area objectives*
 - ❑ *Inform planners on recommended capabilities & investment profile across mission area*
 - **Mission Architecture Studies:**
 - ❑ *Address approaches to meet objectives for single mission*
 - ❑ *Done when little is known of mission & significantly different approaches exist*
 - *e.g., 1st time expedition to study moon of Saturn*
 - ❑ *Scope narrower & deeper than MAA*
 - ❑ *Inform planners on most cost effective approach for mission*



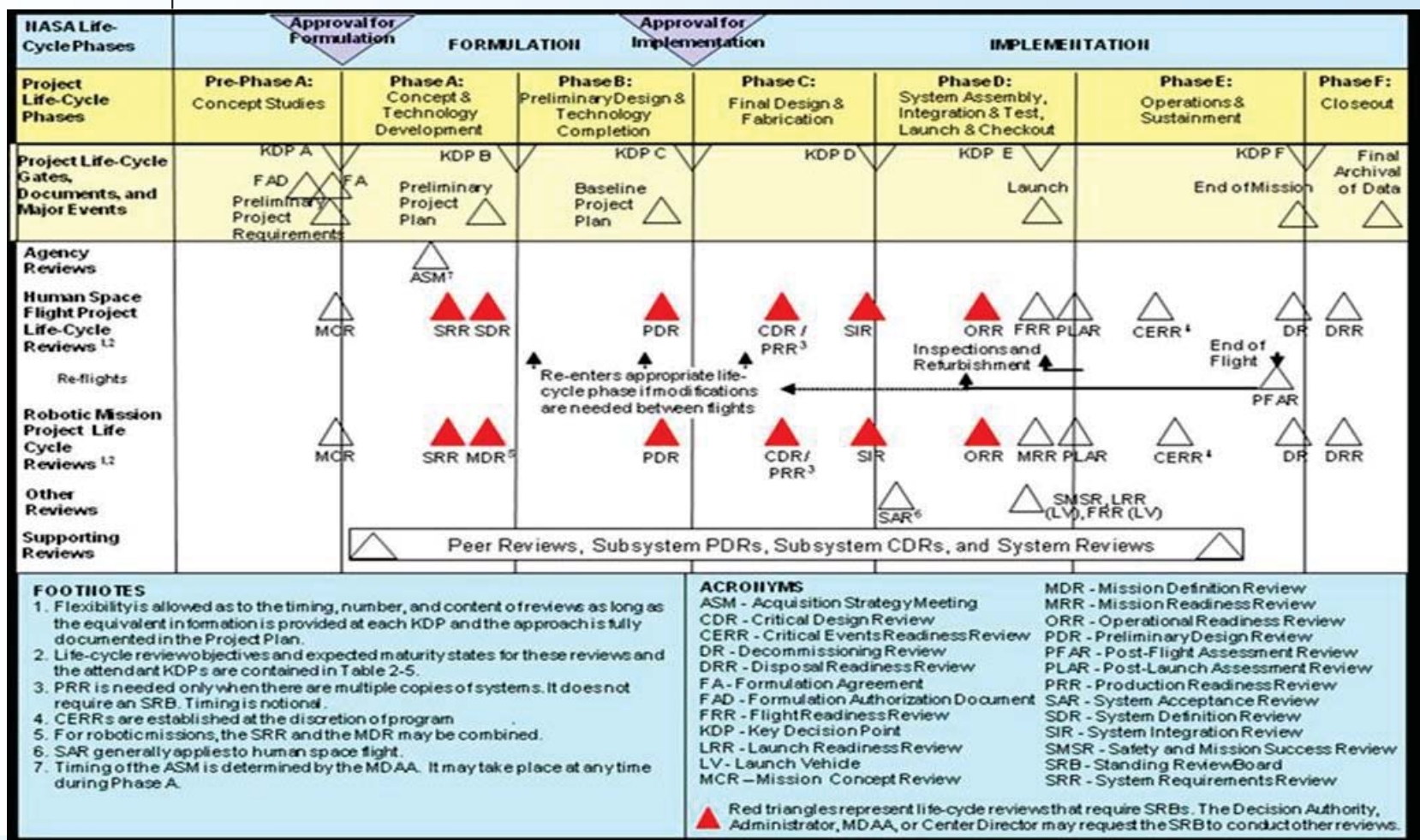


Architecture Development Precedes System Level Concept Design Studies in Project Life Cycle

Figure 1

Architecture Development

NASA Project Life Cycle Example - NASA Procedural Requirements (NPR) 7120.5E



FOOTNOTES

- Flexibility is allowed as to the timing, number, and content of reviews as long as the equivalent information is provided at each KDP and the approach is fully documented in the Project Plan.
- Life-cycle review objectives and expected maturity states for these reviews and the attendant KDPs are contained in Table 2-5.
- PRR is needed only when there are multiple copies of systems. It does not require an SRB. Timing is notional.
- CERRs are established at the discretion of program.
- For robotic missions, the SRR and the MDR may be combined.
- SAR generally applies to human space flight.
- Timing of the ASM is determined by the MDAA. It may take place at any time during Phase A.

ACRONYMS

- ASM - Acquisition Strategy Meeting
- CDR - Critical Design Review
- CERR - Critical Events Readiness Review
- DR - Decommissioning Review
- DRR - Disposal Readiness Review
- FA - Formulation Agreement
- FAD - Formulation Authorization Document
- FRR - Flight Readiness Review
- LRR - Launch Readiness Review
- LV - Launch Vehicle
- MCR - Mission Concept Review
- MDR - Mission Definition Review
- MRR - Mission Readiness Review
- ORR - Operational Readiness Review
- PDR - Preliminary Design Review
- PFAR - Post-Flight Assessment Review
- PLAR - Post-Launch Assessment Review
- PRR - Production Readiness Review
- SAR - System Acceptance Review
- SDR - System Definition Review
- SIR - System Integration Review
- SMSR - Safety and Mission Success Review
- SRB - Standing Review Board
- SRR - System Requirements Review

Red triangles represent life-cycle reviews that require SRBs. The Decision Authority, Administrator, MDAA, or Center Director may request the SRB to conduct other reviews.





Architecture Frameworks

- **Many architecture frameworks reported developed or in use**
 - *A survey of over 60 frameworks is at ref. (a), including those for:*
 - *Enterprise, defense, information, software, automotive, business, security, etc.*





Objective 1

- ***Describe what an MAA is and show how it integrates into the NCS architecture framework***





Beginning Definitions

- ***Before getting started, just what is an “architecture”?***





Beginning Definitions (Cont'd)

- ***New Webster Dictionary (1975) defines “Architecture” as:***
 - 1) the art or science of building; specif. the art or practice of designing and building structures and esp. habitable ones*
 - 2) formation or construction as, or as if, the result of conscious act*
 - 3) architectural product or work*
 - 4) a method or style of building*

- ***New Webster Dictionary (1975) defines “Architect” (from Latin “architectus”, from Greek: “architekton” or master builder) as:***
 - 1) one who designs buildings & superintends their construction*
 - 2) one who plans and achieves a difficult objective (e.g., a military victory)*





What is the “NCS Architecture”?

- ***From these definitions, it’s clear architecting involves some level of design & orchestration, but***
 - *What level of design?*
 - *What does an architecture look like, and what does it do?*
- ***To answer these questions, we’ll need a common view of:***
 - *NCS Architecture & its constituent MAAs*
 - *Core elements of NCS architecture*





Core Elements of an NCS Architecture

- 1) The set of **functional capabilities** that characterizes actual or forecast capabilities of NCS physical assets & human command & control (C2) entities
 - Includes “what” capability will be delivered along with measures of performance (MOPs), e.g.,
 - Quality, quantity, timeliness, interoperability, & robustness (QQTIR)
(Note: this is not an exclusive list)
- 2) The set of NCS **physical assets** (hardware/software) that is, (or is forecast to be) available along with their interconnectivities
 - Shows “how” architecture functional capabilities will be delivered
- 3) The set of NCS **human C2** operator / decision maker **entities** available along with their interconnectivities
 - Note: Automated C2 assets are considered part of physical assets





Core Elements of an NCS Architecture (Cont'd)

- 4) *The concept of operations (**CONOPS**) that identifies how NCS physical assets & human C2 entities will be employed in time sequence to meet a defined mission*
 - *Used to evaluate effectiveness, etc., as function of environment & scenario*

- 5) *The set of **constraints** , i.e., rules / policies & standards / protocols, that constrain use of NCS assets & human C2 entities*
 - ***Each element above pertains to specific period in time, or “epoch”***





NCS Architecture Framework Example

- ***Framework is established by functional decomposition***
 - *Standard systems engineering (SE) technique*
- ***Enables means to identify***
 - *Vertical flowdown of guidance*
 - *Horizontal interfaces within & among architectures*

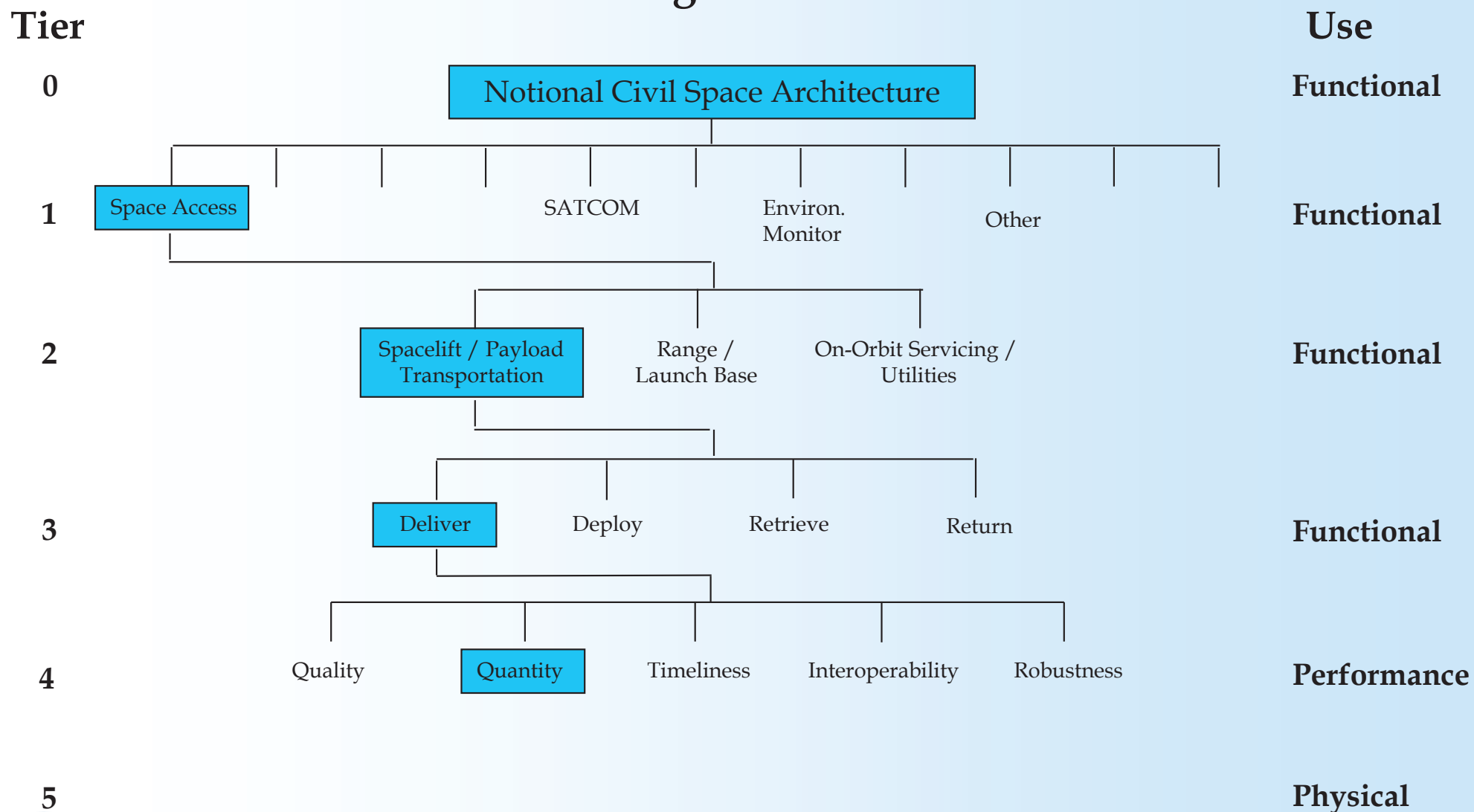




NCS Architecture Framework Example

Space Access Mission Area (Epoch = 20xx)

Figure 2





Functional Decomposition Example

Space Access Mission Area (Epoch = 20xx)

- **Tier 0:** NCS architecture functions applicable to all mission areas
- **Tier 1:** Allocates Tier 0 functions to mission areas, e.g., provide **Space Access**
- **Tier 2:** Allocates Tier 1 functions to sub mission area functions (e.g., provide **Spacelift / Payload Transportation**, etc.)
- **Tier 3:** Allocates Tier 2 functions to more detailed functions (e.g., **deliver, deploy, retrieve, return**, etc.)
- **Tier 4:** Allocates Tier 3 functions to metrics (**QQTIR**) & **MOPs**, e.g., for “**deliver**” function
 - Example quantity metric = x payloads of y,000 kg to z,000 km circular orbit at i° inclination
 - Example MOP = 2 payloads of 2,000 kg to 400 km circular orbit at 51.6° inclination (adds specific values)
- **Tier 5:** Allocates Tier 4 to physical assets & human C2 entities

Note: Number of tiers can vary among mission areas





Role of Tier 0 & 1 Guidance

- **Tier 0: Provides guidance for all mission areas, e.g.,**
 - *Environmental (e.g., power / fuel sources, orbital debris, planetary protection, etc.) policy*
 - *Interoperability standards*
 - *Criticality categories which drive level of robustness (or fault tolerance needed); might pertain to those that assure:*
 - 1) Human survival*
 - 2) Specific mission operational capabilities*
 - 3) Specific technology capabilities*
- **Tier 1: Adds guidance unique to each Tier 1 mission area**
- **Note:**
 - *A fault means loss of capability for any reason (component failure, hostile action, etc.)*
 - Severity of potential fault depends on severity of threat*





Functional Decomposition Table Example

Space Access Mission Area (Epoch = 20xx)

Supports Figure 2

Table 1

Tier 0	Tier 1	Tier 2	Tier 3	Tier 4	Notes
1.0 Provide NCS capabilities	1.1 Provide Space Access capabilities	1.1.1 Provide Spacelift / Payload Transportation capabilities	1.1.1.1 Provide capability to deliver payload(s) to orbit	1.1.1.1.1 Quality	
				1.1.1.1.2 Quantity	
				1.1.1.1.3 Timeliness	
				1.1.1.1.4 Interoperability	
				1.1.1.1.5 Robustness	
			1.1.1.2 Provide capability to deploy payload(s) on orbit	1.1.1.2.1 Quality	
				1.1.1.2.2 Quantity	
				1.1.1.2.3 Timeliness	
				1.1.1.2.4 Interoperability	
				1.1.1.2.5 Robustness	
			1.1.1.3 Provide capability to retrieve payload(s) on orbit	1.1.1.3.1 Quality	
				1.1.1.3.2 Quantity	
				1.1.1.3.3 Timeliness	
				1.1.1.3.4 Interoperability	
				1.1.1.3.5 Robustness	
			1.1.1.4 Provide capability to return payload(s) from orbit	1.1.1.4.1 Quality	
				1.1.1.4.2 Quantity	
				1.1.1.4.3 Timeliness	
				1.1.1.4.4 Interoperability	
				1.1.1.4.5 Robustness	
		1.1.2 Provide Range / Launch Base Capabilities	Use construct similar to 1.1.1	Use construct similar to 1.1.1	
		1.1.3 Provide On-Orbit Servicing / Utilities Capabilities	Use construct similar to 1.1.1	Use construct similar to 1.1.1	



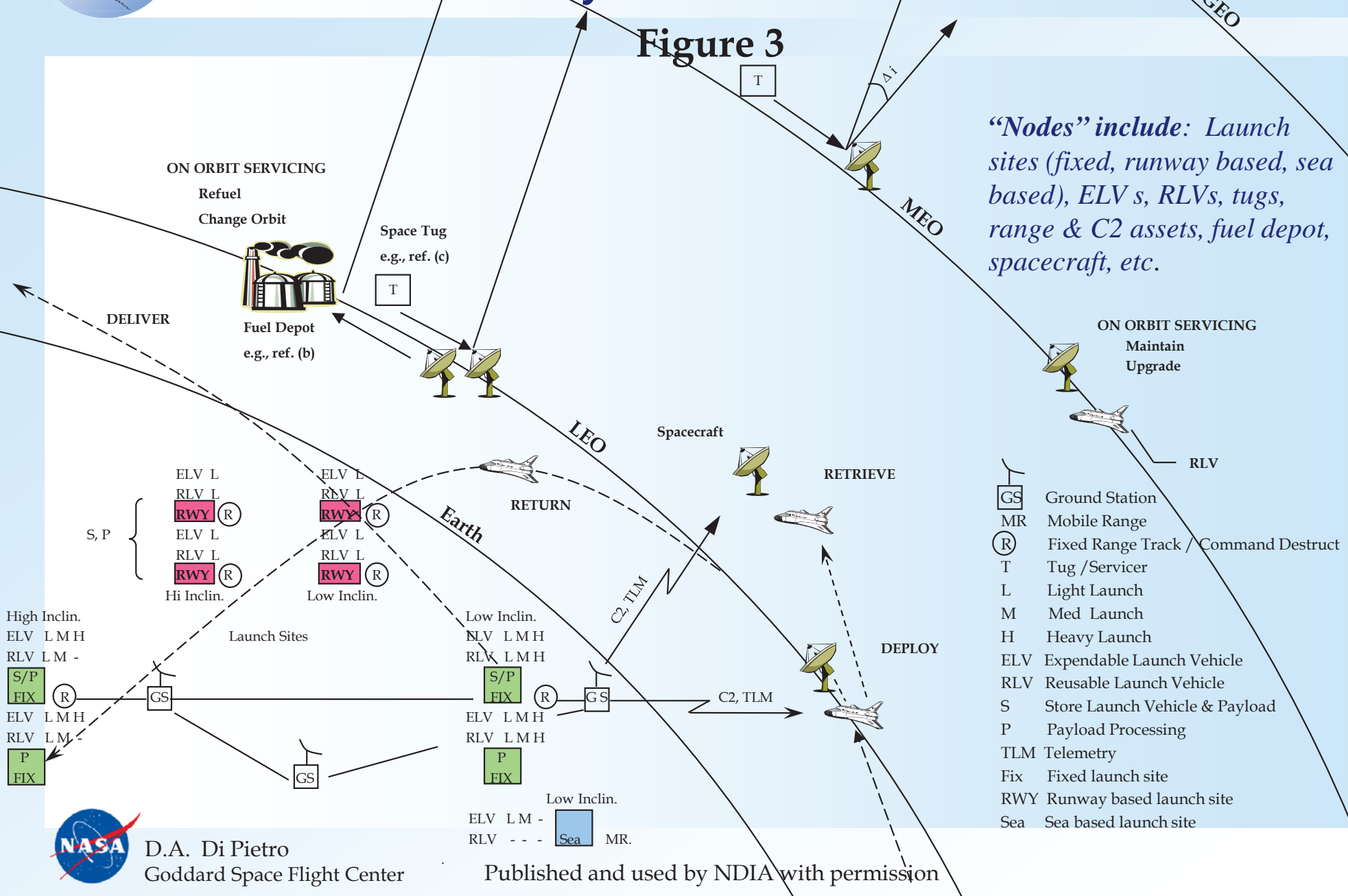


NCS Space Access Physical View (20xx)

Addresses Functional Decomposition

Excludes Assets beyond GEO

Figure 3



“Nodes” include: Launch sites (fixed, runway based, sea based), ELV s, RLVs, tugs, range & C2 assets, fuel depot, spacecraft, etc.





NCS Architecture Framework Example

Mission Area Functions Allocated to Multiple Organizations

Figure 4

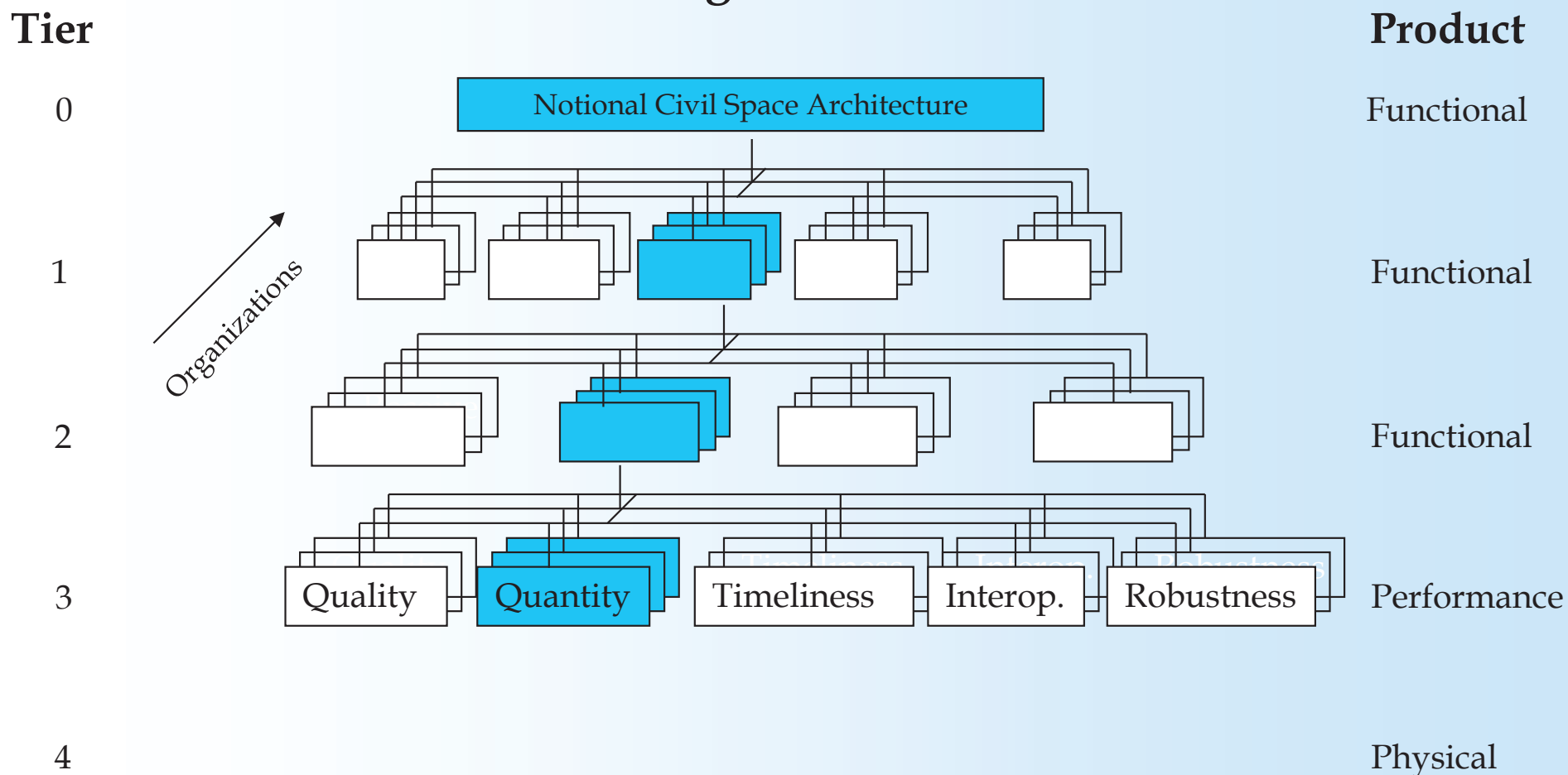


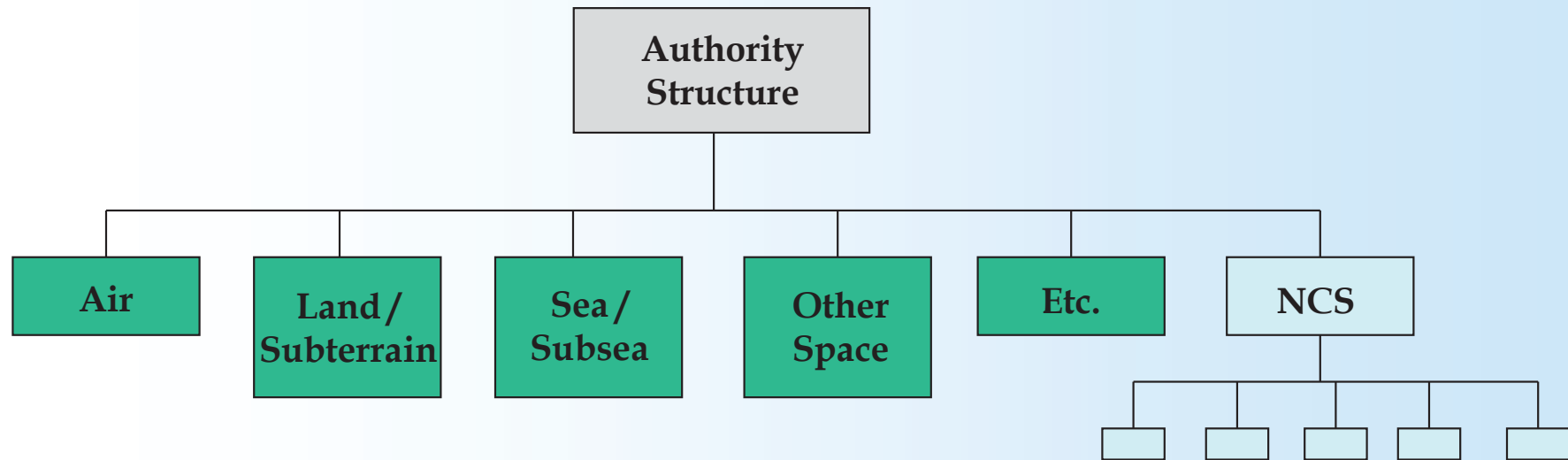
Figure 4 illustrates an NCS architecture framework wherein mission area functions are performed by more than one civil organization





NCS: Just One Domain of Many Domains

Figure 5



- *NCS architecture may be part of larger architecture that crosses domains & stakeholders*
- *Integration with larger architecture may impose additional constraints*





Example Level of “Design” Work in MAA Development

- ***MAA technical analysis typically limited to 1st principles***
- ***For space access MAA with tugs that maneuver spacecraft, ADT might size tugs at rocket equation level***
 - *Tug mass might scale to 1st order via rocket equation & other relationships, e.g., dry mass to propellant mass ratio, etc.,*
- ***No detailed tug subsystem design conducted***





Measures Of Effectiveness (MOEs)

- **MOEs - typically address effectiveness at architecture level & differ from MOPs, e.g.,**
 - *MOP might pertain to sizing nodes for spacelift, range, & on-orbit servicing functions*
 - *MOE might pertain to how well these nodes combine to meet an operational objective of a scenario at MAA level*
- **MOEs typically need to be decomposed into measurable terms in order to be useable by ADT**
 - *Need early & continued customer / user engagement to develop & refine*





Architecture Scenarios & Environments

- **Scenarios**
 - *Include driving operational cases at architecture level*
- **Environments typically are assumed conditions in which architecture will be developed & / or operate, e.g.,**
 - *Stable / cooperative vs. unstable / uncooperative governments*
 - *Stable vs. unstable budgets*
 - *Contested vs. uncontested space operations*
 - *Orbital debris / space weather, etc.*
- **Key enabler for NCS architecture level effectiveness analysis**
 - *Consistent scenarios & environments at MAA & NCS levels for given epoch*





Interface Identification

- ***Horizontal interfaces (within or among MAAs) can be highlighted on functional decomposition***
 - *e.g., transmit data rate / frequency from remote sensing node (Environmental Monitoring MAA) to ground station (SATCOM MAA)*
- ***Some physical interfaces may need to be standardized***
 - *e.g., for some on-orbit servicing nodes*
- ***Horizontal integration analyses across MAAs validate interfaces are compatible***





Mission Area CONOPS Development & Use

- **Each MAA has CONOPS that applies to particular scenarios, environments & epoch**
 - *Used to evaluate MAA effectiveness*
- **CONOPS is specific to architecture design**
 - *i.e., scenario is met differently by space access MAA having RLVs & on-orbit servicing than by MAA having only ELVs*
 - ❑ *RLV = Reusable launch vehicle*
 - ❑ *ELV = Expendable launch vehicle*





Some Uses for an NCS Architecture Framework

- ***Establishes common lexicon for functions, metrics, & products***
- ***Allows synthesis of Tier 0 Architecture (as-is, to-be, should-be)***
 - *Provides coherent context & relationships among architecture elements*
- ***Provides structured flowdown of policy & guidance into MAAs***
- ***Highlights whether studies are for:***
 - *a) One mission area across all QQTIR metrics*
 - *b) All mission areas for only one metric, e.g., timeliness*
- ***Facilitates horizontal (& cross organizational) integration of MAA interfaces at NCS level***
- ***Exposes gaps / overlaps that can be used to select follow-on architecture studies***
- ***Facilitates identifying Tier 0 CONOPS & conducting effectiveness analysis of Tier 0 Architecture in set of scenarios & environments***





Objective 2

- ***Describe an effective approach for developing an MAA***





Terms of Reference (TOR)

- **TOR is top level study charter, approved by customer**
 - *ADT leadership may assist in developing*
- **TOR should clearly identify**
 - *Who, what, where, why, when of study process & products*
 - ❑ *Incl. resources, participants, roles & responsibilities*
- **TOR typically will include**
 - *Problem background (incl. relationship to relevant past studies)*
 - *Problem statement: Concise & clear*
 - *Study scope & product depth, i.e.,*
 - ❑ *Functional boundaries (e.g., include spacelift, exclude on-orbit servicing)*
 - ❑ *Stakeholders*
 - ❑ *Domains*
 - ❑ *Epoch*
 - ❑ *Mission area guidance (e.g., relevant policy directives, etc.)*
 - *Guidance for establishing MOEs*
 - *Definitions for key unique terms*





Terms of Reference (TOR) (Cont'd)

- *Assumptions, Constraints, Groundrules*
 - ❑ *System (x) from stakeholder (y) is out of scope*
 - ❑ *Use data from source (z) as principal input*
 - ❑ *Scenarios & environments*
 - ❑ *Technology readiness date*
 - ❑ *Policy, Cost*

- *Guidance on how to select recommended architecture*
 - ❑ *e.g., single best value architecture within cost constraint, etc.*

- ***TORs are deceptively difficult, but worth time to develop well***
 - *Meaningful TOR can save ADT's significant time*
 - *Weak TOR can leave ADT to define purpose, scope, deliverables, etc. while designing MAA*
 - ❑ *ADT view may not match customer view*





Mission, Domain, & Stakeholder Space

Figure 6

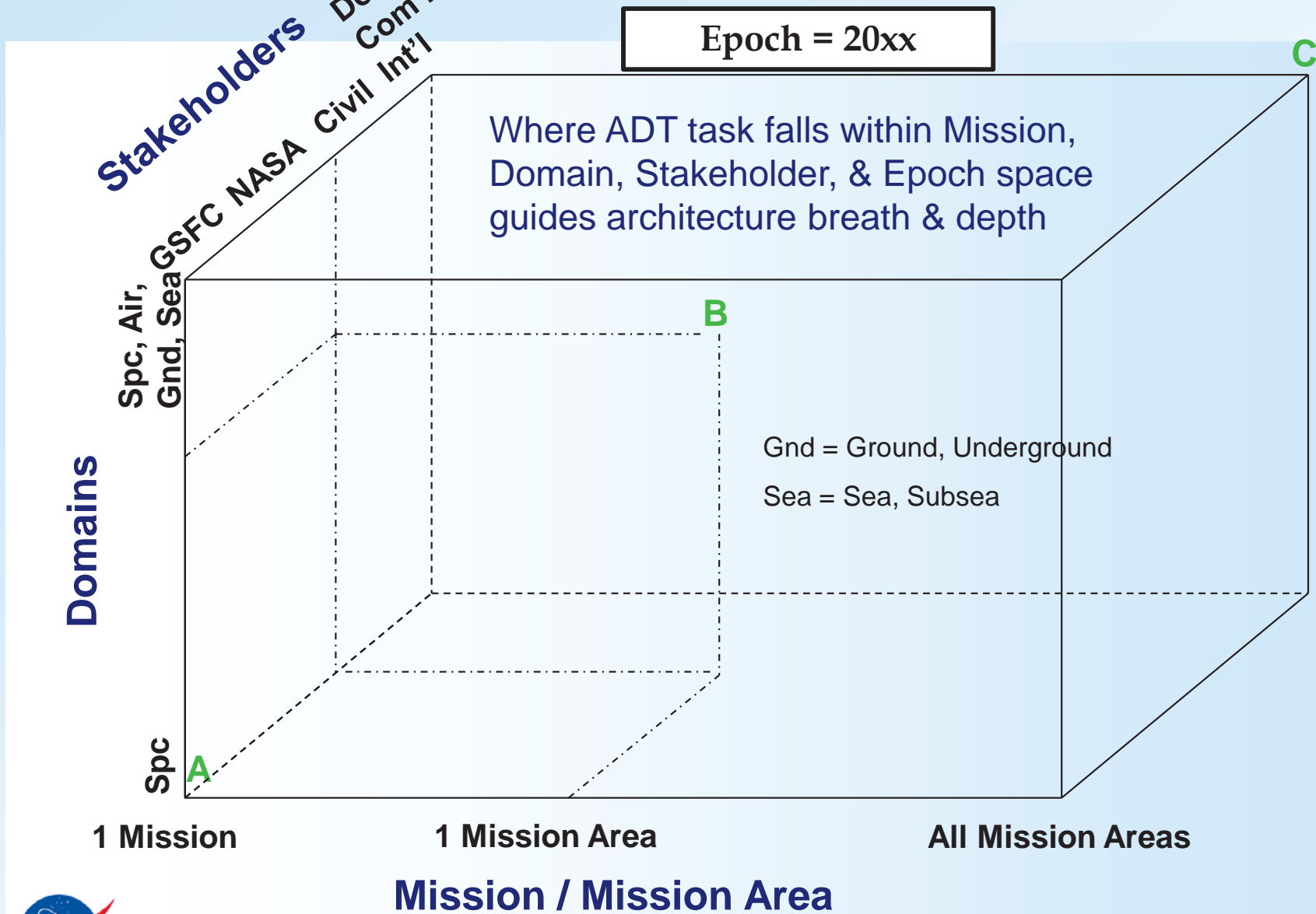
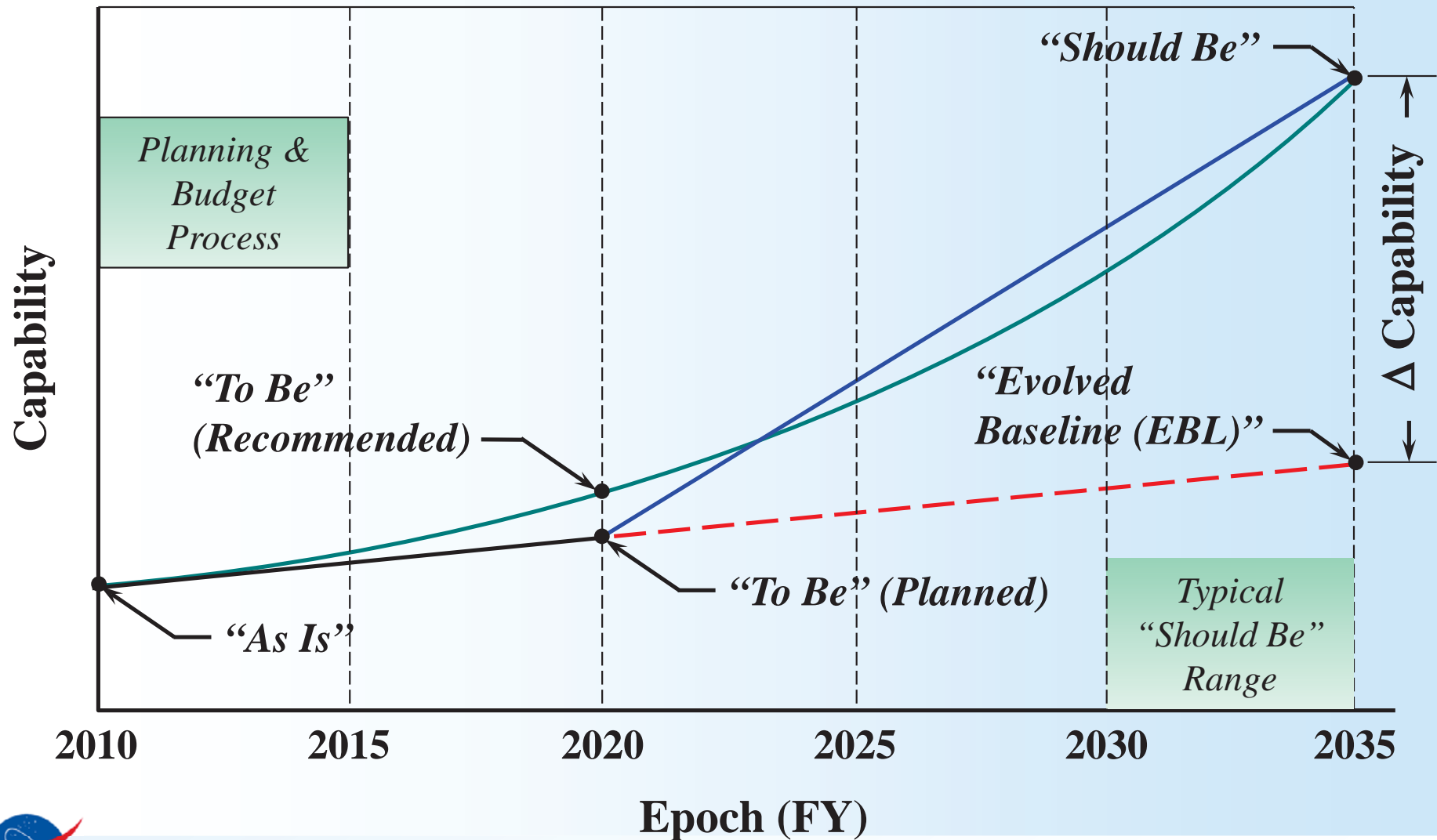




Illustration of “As-Is”, “To-Be”, “Should-Be”, & Evolved Baseline” Architectures

Ref. (d)

Figure 7





Conducting Effective Architecture Studies

- **Lets now look at one way to effectively conduct an MAA study**
 - *A generic, iterative “design cycle” process*
- **Important Note:**
 - *MAA studies can be conducted more than one way*
 - ❑ *Typically, however, they involve synchronized, iterative process with active systems engineering (SE) leadership*
 - ❑ *“Art” is in enabling & orchestrating highly dynamic & creative efforts to produce products of sufficient scope & fidelity within allotted time*





Introduction to Design Cycle Process for Architecture Studies

- **Design cycle process is structured, iterative approach**
 - *Based on standard SE technique for conducting requirements development, design, & analysis*
 - *Brings products to common, coherent reference point in each cycle*
 - ❑ *Accelerates start of architecture design, surfaces unknowns early*
 - ❑ *Provides discrete opportunities for stakeholder / management review*
 - ❑ *Maintains synchronization of assumptions, trades & analyses*
 - ❑ *Facilitates systems level integration*
 - ❑ *Improves final report & reduces work required to produce it*
- **Other process models (e.g., waterfall, ad-hoc iterative, etc.), less effective for studies with high uncertainty**
 - *Waterfall (i.e., linear, unidirectional) processes more effective for tasks that are well understood*
 - *Ad-hoc iterative processes difficult to keep in synch*





Introduction to Design Cycle Process for Architecture Studies (Cont'd)

- ***First time MAA developments are inherently exploratory & uncertain***
 - *Teams learn at high rate, unknown-unknowns (UUs) dominate early*
- ***Can't plan all study details at outset***
 - *Outline general plan (incl. major activities & milestones) early*
 - *Plan cycle details iteratively within general plan constraints*
 - ☐ *Plan & execute Cycle 1, plan & execute Cycle 2a, etc.*
- ***Starting design work early accelerates learning***
 - *Helps surface UUs early*
 - *Allows adjustments when there is still time to resolve*





Design Cycle Approach Overview Conducted in 3 Cycles

- ***Cycle 1: Pathfinder; learn & assess readiness for design***
 - ❑ *a) requirements characterized in form usable for analysis*
 - ❑ *b) metrics compatible with modeling tools*
 - ❑ *c) modeling tools can analyze design to provide desired product set*
 - ❑ *d) desired product set suffices to answer questions in TOR*
 - *Analyze a few architectures that span solution space*
 - *Surrogates can be used for requirements, technology forecast*

- ***Cycles 2a & 2b:***
 - *Conduct comprehensive investigations for broad range of candidate architectures*
 - *Determine most promising architectures across trade space*

- ***Cycle 3:***
 - *Refine designs & analyses on most promising representative architectures of solution space*
 - *Recommend single best-value architecture*

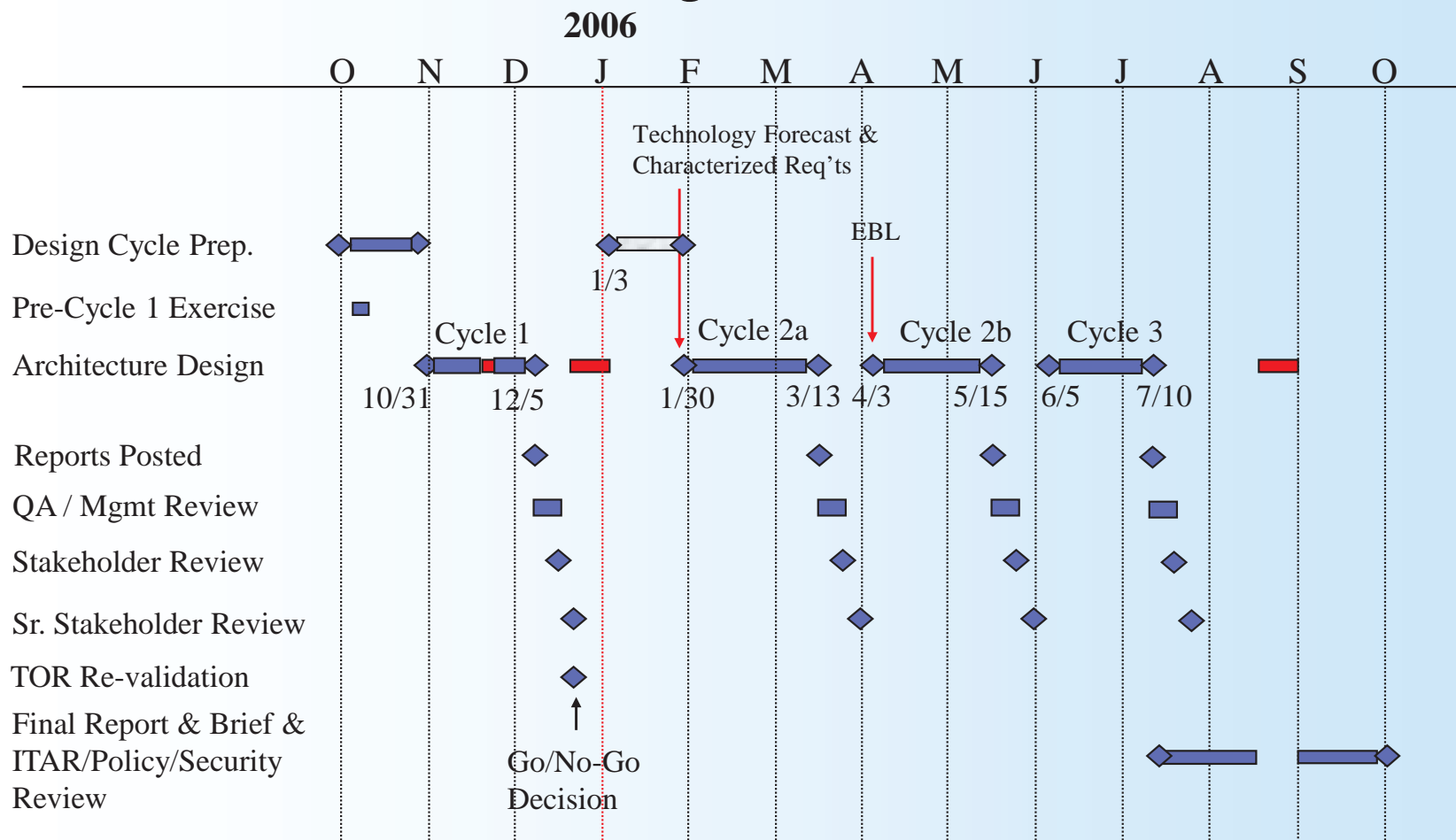




12-Month MAA Study Design Cycle Template

CY 2005/2006 Example with Pre-Design Products Available

Figure 8



- 1) Assume no scheduled work during: 1) Thanksgiving week, 2) last 2 weeks of August & December. Assume partial week during Spring Break
- 2) End of Cycle 1 presents opportunity to assess whether study should be continued (e.g., is problem well posed, is scope realistic, etc.) and to revalidate TOR.





Pre-Design Products

Draft Products Developed before Cycle 1

- **Pre-Design Products Accelerate Cycle 1 start**
 - *Functional decomposition through performance metrics*
 - *Generic scalable physical nodes*
 - *Prepare for modeling use, incl. governing equations / relationships*
 - *Generic “threads” (see next chart)*
 - *Types of modeling tools available to analyze nodes*

 - *Technology forecast (to degree readily available in roadmaps, etc.)*
 - *Summary of known mission area guidance & relevant studies*
 - *MOEs previously used or identified for mission area*

- **Pre-design products may also include**
 - *Data collection templates that support development of technology forecast and as-is, to-be planned, and EBL architectures*





Architecture Trade Case Matrix

Space Access Example

- **Analyses of individual nodes combine to determine performance / effectiveness of “threads”**
 - *Threads contain all elements to deliver an end-to-end service, e.g.,*
 - ❑ *Deliver payload to orbit includes: launch base, ground station, range, launch vehicle, human C2 entities*
 - *Nodes are main elements (e.g., launch vehicle) within thread*
- **Analyses of individual “threads” combine to determine performance / effectiveness of MAA**
 - *ADTs assign combinations of threads to a range of candidate MAAs*
- **Functional decomposition for final MAA solution transferred into NCS functional decomposition table**
 - *Formats similar*





Architecture Trade Case Matrix

Leverages Functional Decomposition Table

Space Access Example

Table 2

Architecture Solution (How's) =>	1a	1b	1c	2a	2b	2c	3a	3b	3c	4a	4b	4c	5a	5b	5c	6a	6b	6c
Functions / MOPs (What's)																		
PROVIDE Space Access Capabilities	All ELV			Mix ELV / RLV			All RLV w/Tugs											
Provide Spacelift / Payload Transportation Capabilities																		
- Deliver																		
- Quality										<p>Each “architecture” is a composite of several “threads” designed to meet MOPs (QQTIR)</p> <p>Architecture 1 represents an all ELV solution where threads 1a, 1b, & 1c might be light, medium, & heavy ELVs, respectively.</p> <p>Architecture #3 represents an all RLV solution with tugs, where threads 3a, 3b, & 3c might be light RLVs, medium RLVs, & medium tugs, respectively.</p>								
- Quantity																		
- Timeliness																		
- Interoperability																		
- Robustness																		
- Deploy (QQTIR as above)																		
- Retrieve (QQTIR as above)																		
- Return (QQTIR as above, etc.)																		
Provide Range / Launch Base Capabilities										Expand as done for Spacelift / Payload Transportation								
Provide On-Orbit Servicing / Utilities Capabilities										Expand as done for Spacelift / Payload Transportation								

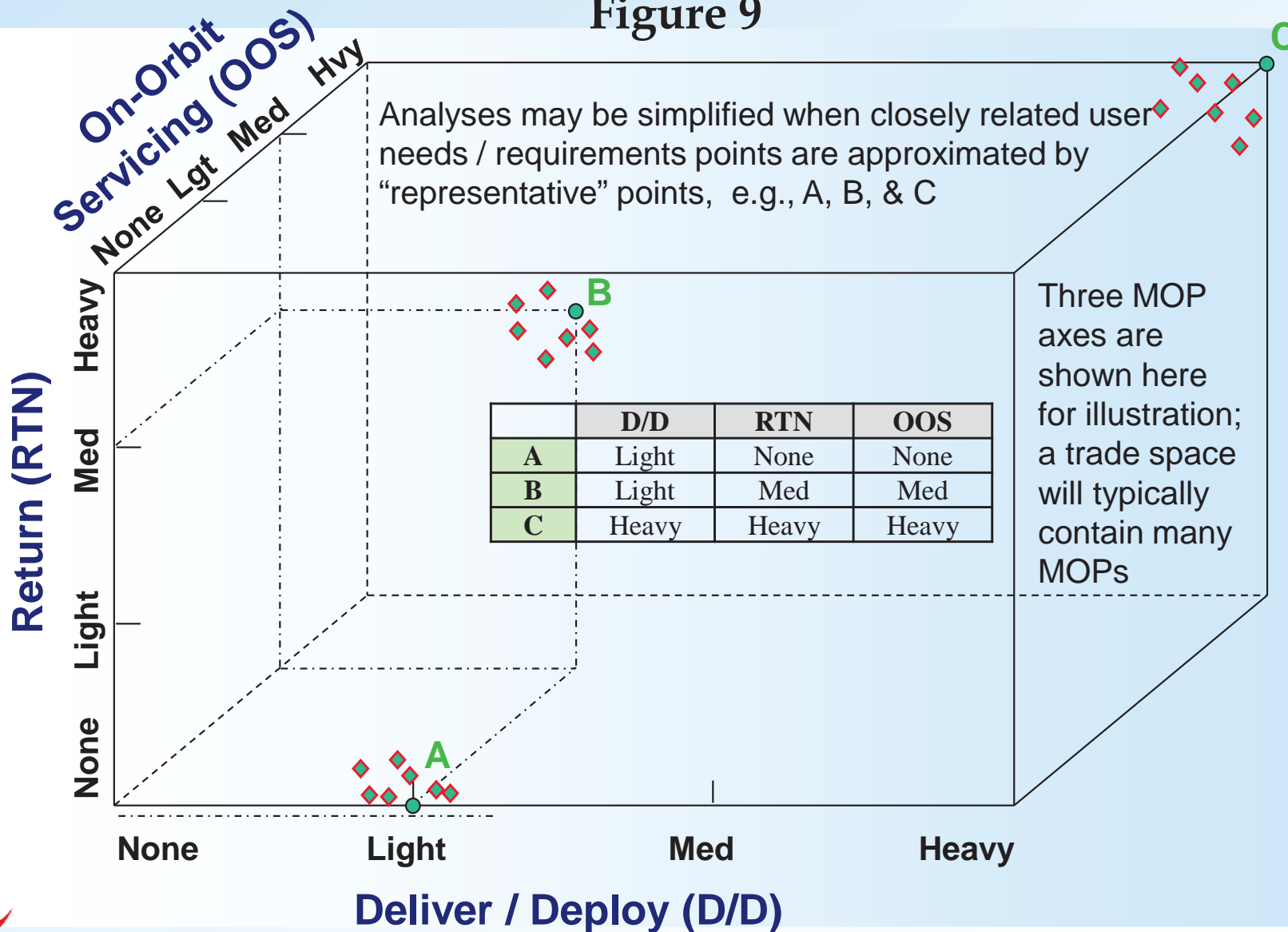




Example Requirements Trade Space

Space Access Example

Figure 9





Typical Design Cycle Products (1 of 2)

Table 3

No	Product	Cycles
1	Functional Decomposition (& MOPs / Interface Req'ts)	1, 2a, 2b, 3
2	Needs / Requirements Classes & Bounding Cases	*, 2a, 2b, 3
3	Trades and Tradespace Report	1, 2a, 2b, 3
4	EBL	2b, 3
5	Technology Forecast & Plan	*, 2a, 2b, 3
6	Architecture Alternative (AA) Point Designs	1, 2a, 2b, 3
7	Scenarios	1, 2a, 2b, 3
8	Threat / Alternative Future Environments	1, 2a, 2b, 3
9	MOEs	1, 2a, 2b, 3

Note: Shading aggregates products into ADT subteam reports

- 1) Operations: Green shading
- 2) Systems: Blue shading
- 3) Analysis: Yellow shading
- 4) Architecture SE: Grey shading

* Surrogates may be used for Cycle 1





Typical Design Cycle Products (2 of 2)

Table 3 (Cont'd)

No	Product	Cycles
10	Performance / Utility Analyses Report	1, 2a, 2b, 3
11	CONOPS	1, 2a, 2b, 3
12	Vulnerability Assessment	1, 2a, 2b, 3
13	Doctrine / Policy Assessment	1, 2a, 2b, 3
14	Work Breakdown Structure	1, 2a, 2b, 3
15	Cost Analysis	1, 2a, 2b, 3
16	Risk Assessment	1, 2a, 2b, 3
17	Subteam Technical Reports	1, 2a, 2b, 3
18	Systems Engineer Report	1, 2a, 2b, 3





Some Recommended Practices for MAA Development

- **Develop Strong TOR**
 - *Sound understanding of objectives, scope, product, etc.*
- **Set “Should-Be” epoch far enough out for candid discussion**
 - *25 years: Allows candid discussion of future architecture*
 - *15 years: Discussion highly constrained by current budget*
- **Conduct ADT in cycles vs. single, waterfall step**
 - *Keep Cycle 1 short, but apply concerted effort – high value learning*
 - *Avoid pressure to use results from Cycle 1 for budget inputs*
- **Don’t retrofit architectures from prior cycles**
 - *Just apply what’s been learned to future cycles*
- **Exercise full solution space in Cycles 1, 2a & 2b**
 - *Cycle 1 will have few architectures, but will span solution space to exercise thought paths*





Some Recommended Practices for MAA Development (Cont'd)

- ***Engage stakeholders with coherent products periodically***
 - *Don't wait until end of study to engage*
- ***Begin writing ADT report in Cycle 1, refine in Cycles 2 & 3***
 - *Write reports first (documents of record), then translate to briefings*
- ***Assign architecture systems engineer experienced in successfully conducting MAA studies***
- ***Remain objective & impartial***
 - *Recognize unknown unknowns dominate early*
- ***Respect the clock***





Questions ?





References

- a) ***<http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html>***
- b) ***http://en.wikipedia.org/wiki/Propellant_depot***
- c) ***http://en.wikipedia.org/wiki/Space_tug***
- d) ***Adapted from model used by Mr. H. E. Hagemeyer, Deputy Director, National Security Space Office, 2009***





Backup





Use of the Term “Requirements” for MAAs

- ***“Requirements” as used by an ADT have a different context than in project management***
 - *An ADT uses “requirements” to reflect classes of user needs while conducting MAA trade study cases prior to Pre-Phase A*
 - *In project management, “requirements” aren’t baselined until System Requirements Review (SRR), normally near the end of Phase A, for a specific mission concept*





NCS CONOPS Development & Use

- ***At NCS level, a CONOPS is developed using scenarios, environments, & epoch consistent with those evaluated for each MAA***
 - *Assumes MAA's within NCS have significant operational interrelationships*
- ***Effectiveness of NCS architecture is periodically evaluated using this CONOPS with “frozen” architecture design***
- ***Output of effectiveness assessment highlights parts of NCS architecture that underperform either due to shortfall in capability or due to interface incompatibility***
 - *These areas may be candidates for study in next cycle of NCS architecture design*





Current, Mid-Term, & Far Term Architectures, Detail for Figure 7

Ref. (d)

- **Architectures are developed for three periods in time, current, mid-term, & far-term**
 - *Current architecture is referred to as “**as-is**” architecture*
 - *Mid-term architecture is referred to as “**to-be**” architecture*
 - *Far-term architecture is referred to as “**should-be**” architecture*
- **Figure 7 shows these architectures; where “as is” is FY 2010**
 - **“Should-be” architecture**
 - ❑ *Associated with systems & capabilities determined by ADT to be needed in “**should be**” epoch of FY 2035*
 - **“To-be planned” architecture**
 - ❑ *Associated with systems & capabilities that would result from proceeding on current development path (i.e., includes efforts funded in any year of current budget) until “to-be” epoch of FY 2020*





Current, Mid-Term, & Far Term Architectures, Detail for Figure 7 (Cont'd)

Ref. (d)

- **“Evolved baseline”, or EBL**
 - ❑ Linear extrapolation of **“to-be planned”** architecture out to **“should-be”** epoch
 - ❑ Assumes no: non-linear breakthroughs afforded by new technologies, new operational doctrine, new policies, etc.
- **ADT compares “should-be” & EBL architecture capabilities to identify changes needed**
 - ADT translates those changes into capability improvements for **“to-be recommended”** architecture that enables gradual, continuous improvement toward **“should-be”** capability
 - ADT identifies corresponding funding adjustments in budget to meet **“to-be recommended”** capability

