A System Engineering and Acquisition Approach for Space System Software Resiliency

Dewanne Marie Phillips

Software Enterprise Integration Office (SEIO) Lead
The Aerospace Corporation

George Washington University, PhD Candidate

April 11, 2016

Space System Software Resiliency Outline

- Challenges for Space Systems
- Definitions
- Disruptions
- Infrastructure Considerations
- Satellite Systems Survey
- Software Resiliency in Lifecycle
- Acquisition Steps
  - *Technical Software Resiliency Architecture Requirements*
  - *Standards/Policies for Software Resiliency*
  - *Request for Proposal (RFP)/Statement of Work (SOW) for Space Systems*
  - *Software Architecture Resiliency Goals*
  - *Software Architecture Resiliency Quality Attributes*
  - *Architecture Analysis of Alternatives*
- Conclusion

Challenges for Software Intensive Space Satellite Systems

- "National and Departmental level guidance call for bolstering resilience and making resilience a consideration in all architectural planning and evaluation, as well as in all system planning and development activities for DoD space capabilities." --Space Domain Mission Assurance: A Resilience Taxonomy, 2015

- "As we invest in next generation space capabilities and fill gaps in current capabilities, we will include resilience as a key criterion in evaluating alternative architectures….We will develop the most feasible, mission-effective, and fiscally sound mix of these alternatives." --National Security Space Strategy, 2012

- ..."the core of resiliency and resiliency analysis is tied to the overall system engineering of the enterprise." --Northrop Grumman Aerospace Systems, Military Space Resiliency: Definition, Measurement, and Application, Sept 2013

Assertion: Robust software and system engineering practices contribute to the foundation of a resilient software intensive space satellite system

Definition: Resiliency

- “Resilience is the ability of an architecture to support the functions necessary for mission success in spite of hostile action or adverse conditions. An architecture is "more resilient" if it can provide these functions with higher probability, shorter periods of reduced capability, and across a wider range of scenarios, conditions and threats. Resilience may leverage cross-domain or alternative government, commercial, or international capabilities.”

  -- "Memorandum for DoD Executive Agent for Space: Space Resilience Definition and Evaluation Criteria," October 11, 2011.

- ***Software resiliency*** has been defined as, “the ability to reduce the magnitude and/or duration of disruptive events. The effectiveness of a resilient application or infrastructure software depends upon its ability to anticipate, absorb, adapt to, and/or rapidly recover from a potentially disruptive event.”

  -- ”Secure and Resilient Software Development”, 2010.

Software resiliency is essential to achieve a resilient space satellite system

Wide Range of System Faults and Vulnerabilities

- Failure Modes
- Latent vulnerabilities and those introduced through system maintenance and sustainment activities
  - *Training of the User is important for the resiliency of the system*
- Compromises in operational system dependencies
- Reliance on large-scale, widely distributed, network-dependent information systems
- Vulnerabilities (i.e., Malicious)
  - *Internal (Insider)*
  - *External (Adversaries)*
  - *Supply Chain*
- Disruptive Technology

Software engineering must address these faults and vulnerabilities early life cycle

Infrastructure Resiliency Considerations

- Software systems engineering considerations for components/system across the infrastructure must be assessed for operational effectiveness
  – *Potential impacts to the mission (e.g., Data Processing, Command and Control)*
  – *Communications (e.g., GPS signal jamming, encryption, spoofing)*
  – *Loss or Attack on resources (e.g., Systems, Servers, Networks)*
    - Lack of critical services used by the software system
    - Varying requirements among complex composite interactions
    - Inconsistent security controls for peer systems within mission

Software integrity is key to protecting the infrastructure against defects and attacks

Survey of Software Intensive Space Satellite Programs

- Collected data from subject matter experts to answer questions
  - *"Do software resiliency engineering practices have a discernable effect on system resiliency? (Yes/No)"*
  - *"Specifically, did the system remain resilient (Yes/No) as a result of some of the software/systems engineering practices (e.g., standards, requirements, architecture attributes secure design and coding practices and secure test) within the DoD 5000.02 SE life cycle?"*

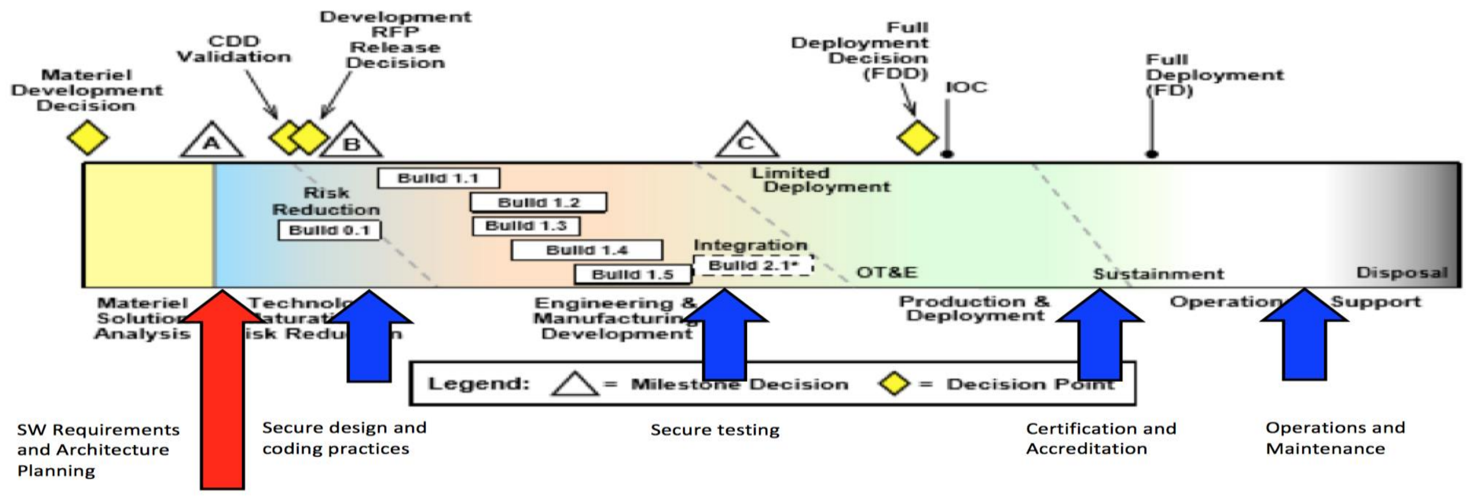
Software Resiliency Data Discussion Sheet (**Y**es / **N**o)

Program	Resilient?	Standards, Policy, Requirements, and Governance	Architecture Quality Attributes							Secure Design and Coding Practices	Secure Testing
			Availability	Redundancy	Survivability	Safety	Security	Reliability	Other		
P1	Y	N	Y	Y (H/W)	Y (H/W)	Y (System)	Y	Y		N	N
P2	Y	Y	N	Y	Y	Y	Y	Y		N	Y
P3	Y	N	N	N	N	N	N	Y		N	N
P4	N	N	Y	N	N	Y	Y	Y		N	N
P5	Y	Y	Y	N	N	N	N	N		N	N
P6	N	N	Y	N	N	Y	N	N		N	N

Software architecture quality attributes contribute to a more resilient software satellite system

Build Software Resiliency Early in the Lifecycle

- Most Software defects are injected early in the life cycle
  - *70% during requirements, architecture, and design*
  - *20% during code and unit test*
  - *10% in Integration*
- Defects decrease the resiliency of the system
- Correcting defects after the design phase typically costs 50-200 times as much as correcting the error in the requirements phase



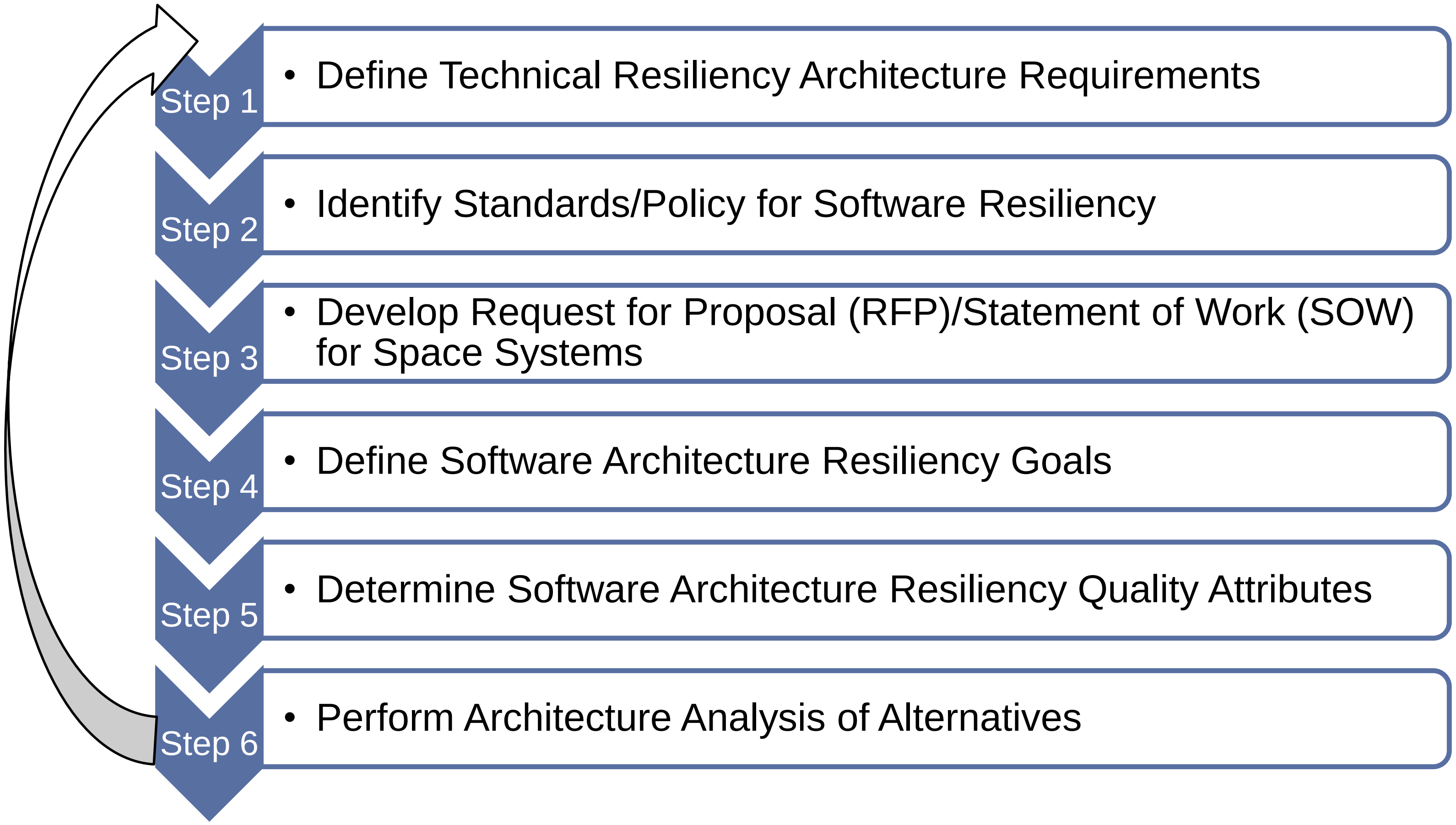
Materiel Development Decision
CDD Validation
Development RFP Release Decision
Full Deployment Decision (FDD)
IOC
Full Deployment (FD)
A
B
C
Build 1.1
Risk Reduction
Build 0.1
Build 1.2
Build 1.3
Build 1.4
Build 1.5
Limited Deployment
Integration
Build 2.1*
OT&E
Sustainment
Disposal
Materiel Solution Analysis
Technology Maturation & Risk Reduction
Engineering & Manufacturing Development
Production & Deployment
Operations & Support
Legend: = Milestone Decision = Decision Point
SW Requirements and Architecture Planning
Secure design and coding practices
Secure testing
Certification and Accreditation
Operations and Maintenance

1

[1] DoD Instruction 5000.02, Figure 4. Model 2: Defense Unique Software Intensive Program, January 7, 2015

Addressing software resiliency requirements early in the SE life cycle will mitigate opportunities for software defects which allow the insertion of threats and vulnerabilities

Recommended Acquisition Steps for a Resiliency Software Architecture



Step 1
- Define Technical Resiliency Architecture Requirements

Step 2
- Identify Standards/Policy for Software Resiliency

Step 3
- Develop Request for Proposal (RFP)/Statement of Work (SOW) for Space Systems

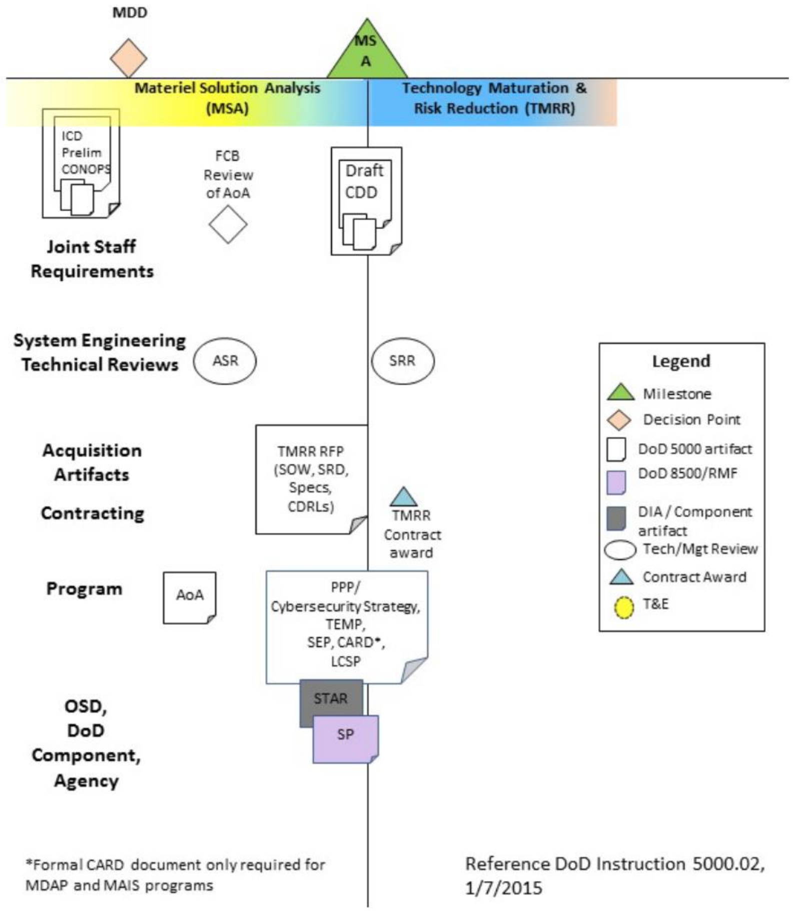
Step 4
- Define Software Architecture Resiliency Goals

Step 5
- Determine Software Architecture Resiliency Quality Attributes

Step 6
- Perform Architecture Analysis of Alternatives

Step 1 - Define Technical Resiliency Architecture Requirements

- During the MSA Phase, software space satellite system architectures define:
  - *Security and resiliency considerations (e.g., Statement of Capability, Concept of Operations)*
  - *Key SW functional/non-functional resiliency requirements*
  - *Software requirements specific to the system that provide protection of essential infrastructure and services*
  - *Major software elements, mission-critical capabilities, and relationships of the system; describe the resiliency characteristics of each identified*

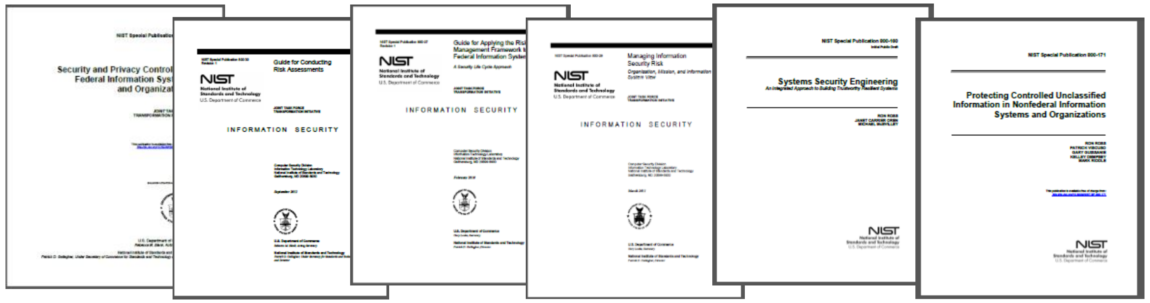


MDD
MS A
Materiel Solution Analysis (MSA)
Technology Maturation & Risk Reduction (TMRR)
ICD Prelim CONOPS
FCB Review of AoA
Draft CDD
Joint Staff Requirements
System Engineering Technical Reviews
ASR
SRR
Acquisition Artifacts
Contracting
TMRR RFP (SOW, SRD, Specs, CDRLs)
TMRR Contract award
Program
AoA
PPP/ Cybersecurity Strategy, TEMP, SEP, CARD*, LCSP
STAR
SP
OSD, DoD Component, Agency
Legend
Milestone
Decision Point
DoD 5000 artifact
DoD 8500/RMF
DIA / Component artifact
Tech/Mgt Review
Contract Award
T&E
*Formal CARD document only required for MDAP and MAIS programs
Reference DoD Instruction 5000.02, 1/7/2015

Software resiliency architecture requirements drive the development, design, and implementation of secure/trusted software intensive satellite systems

Step 2 – Identify Standards/Policy for Software Resiliency

- Review and tailor standards, directives, policies, and controls to address faults and vulnerabilities
  - *National Institute of Standards and Technology (NIST)*

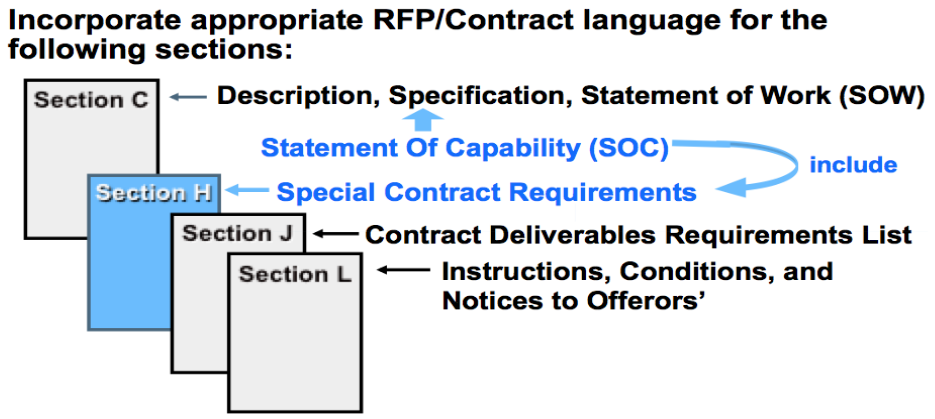

Security and Privacy Control
Federal Information Sys
and Organizat
Guide for Conducting Risk Assessments
INFORMATION SECURITY
Guide for Applying the Risk Management Framework t Federal Information Syste
INFORMATION SECURITY
Managing Information Security Risk
INFORMATION SECURITY
NIST Special Publication 800-160
Systems Security Engineering
An Integrated Approach to Building Trustworthy Resilient Systems
NIST Special Publication 800-171
Protecting Controlled Unclassified Information in Nonfederal Information Systems and Organizations
NIST

  - *Committee on National Security Systems Publications (e.g., CNSSP 12, CNNSI 1200, CNSSI 4009)*
  - *Policies and Guidance (e.g., DoDI 5000.02, 8510, 5200.44, ICD 503)*
  - *Information Security Standards (e.g., ISO/IEC 27001, 27002, 27004)*
  - *Architecture Standards (e.g., ISO/IEC/IEEE 42010, ISO/IEC 42030, 42020)*
  - *Secure Coding Standards (e.g., CERT C++, ISO/IEC TS 17961)*
- Assess DoD guidance to ensure standards are relevant for protecting data from faults and vulnerabilities

Acquisition organizations must strengthen procurement and acquisition standards to produce more resilient systems

Step 3 – Develop Request for Proposal (RFP)/Statement of Work (SOW) for Space Systems

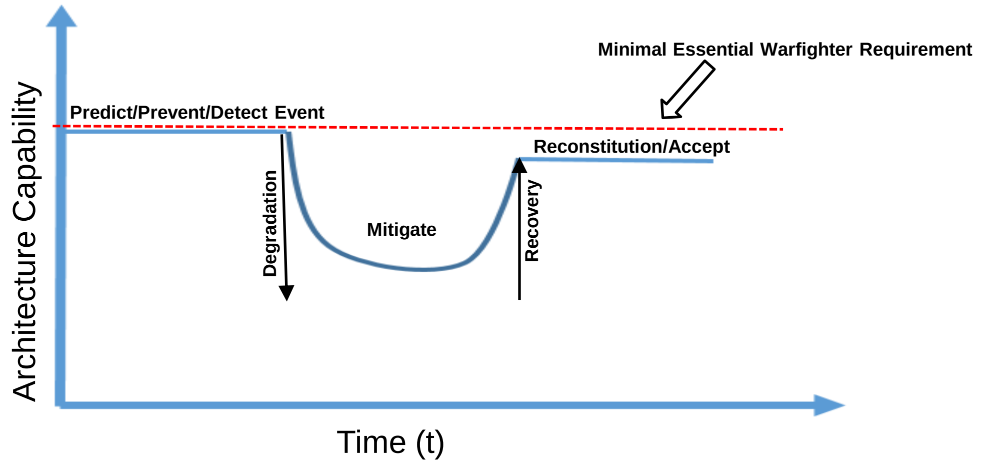
- “Build resiliency into” the architecture and incorporate SW assurance considerations throughout the software acquisition process
- Examples:
  - *RFP/SOW; Software Architecture Quality Attribute Workshop (QAW), Software Architecture Evaluation for Resiliency*
  - *Section J (CDRLs); Test Evaluation Management Plan (TEMP), Architecture Design Document (ADD), Risk Management Plan (RMP), Systems Engineering Plan (SEP), Software Development Plan (SDP)*
  - *Section L; Describe how SW resiliency quality attribute scenarios are integrated into requirements and managed, what are the SW architecture metrics, how are architectural changes managed, how are architectural risks prioritized and mitigated, what are the milestone entry/exit milestone criteria*


Incorporate appropriate RFP/Contract language for the following sections:
Section C
Description, Specification, Statement of Work (SOW)
Statement Of Capability (SOC)
include
Section H
Special Contract Requirements
Section J
Contract Deliverables Requirements List
Section L
Instructions, Conditions, and Notices to Offerors’

Being proactive early in the DoD 5000.02 SE life cycle will help identify and mitigate architectural risks for software intensive satellite systems

Step 4 – Define Software Architecture Resiliency Goals

- (U) Develop questions probing how the architecture will be developed and utilized to recover mission performance in the event of system loss or architecture degradation, such as:
  - *Will the system perform and meet its resiliency goals (e.g., Predict, Detect, Avoid, Mitigate, Reconstitute, Recover) ?*
  - *Can the system withstand a disruptive event with no loss of critical functions?*
  - *Will a loss in a system capability be isolated to prevent it from cascading to other interconnected segments or dependent systems?*
  - *What is the quantitative amount of capability loss tolerable to maintain the mission?*



Example of Software Architecture Resiliency Capability Loss

The software architecture should represent the goals of the stakeholders to define how the system is structured analogous to software resiliency needs

Dewanne.M.Phillips@aero.org

Step 5 – Determine Software Architecture Resiliency Quality Attributes

- Conduct a Quality Attribute Workshop (QAW) (e.g., Robustness, Safety, Reliability, Availability, Survivability, Security, Integrity, Quality, Flexibility)
  - *Present business and/or programmatic drivers for the system and system architectural plans*
  - *Identify Architectural Drivers*
  - *Brainstorm real-world scenarios for the system*
  - *Consolidate scenarios that are similar in content*
  - *Prioritize the scenarios through a voting process*
  - *Refine the top four or five scenarios*

- Examples of key stakeholders software resiliency architectural drivers
  - *Reliable and Secure software – Software development activities provide trustworthy and responsive elucidations that mitigate threats*
  - *Software Recovery – System software must reestablish functionality from anomalies or adverse events, rapidly restore critical operations when attacks are successful*
  - *Advanced Persistent Threat (APT) deterrence – ability to enhance and deploy new software upgrades to system during DoD 5000.02 SE life cycle in response to threats, innovative/disruptive technologies, or other vulnerabilities which may impede mission operations*

Quality attribute requirements stem from business and mission goals; Quality attributes requirements drive the design of the software architecture

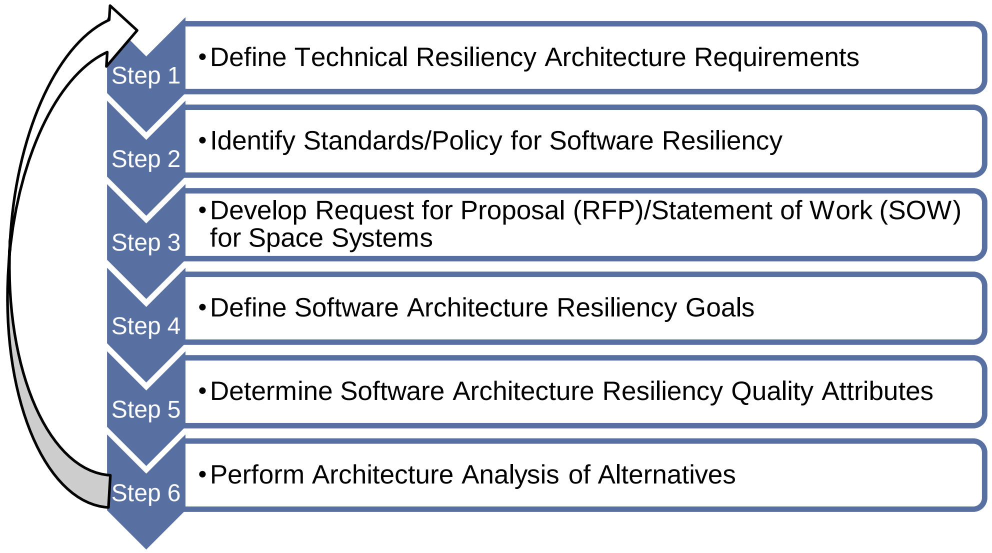
Step 6 – Perform Architecture Analysis of Alternatives

- Define alternative architectures to provide passive resilience and enable protection in depth
- Assess architecture impacts across connected systems - require alignment of risk across all stakeholders and systems , otherwise critical threats will be unaddressed (missed, ignored) at different points in the interactions
- Conduct Architecture Assessments as part of procurement process
  - *The Aerospace Cyber Resiliency Framework provides a methodology to assess the cyber resiliency characteristics of space and ground architectures.* ***[1]***
  - *The Aerospace Software Architecture Evaluation Framework identifies dimensions (e.g., SW resiliency) that address; architecture fundamentals, documentation, functionality, quality attributes, development, and evolution methodologies.* ***[2]***
  - *The SEI Architecture Tradeoff Analysis Method (ATAM) is a methodology to align the interrelationships between a program's key business and mission goals, acquisition strategy, and software architecture that can be used prior to Milestone A or B to increase the probability of program success.*
  - *The MITRE Corporation Framework identifies cyber resiliency goals, objectives, and practices; the threat model for cyber resiliency; architectural layers or domains to which cyber resiliency practices could be applied; and aspects of cost to consider as part of the trade-off analysis for alternative strategies and implementations.*

Existing and next generation software intensive space satellite system architectures must be assessed to mitigate resiliency capability gaps

Conclusion

- Six steps to a more resilient system


Step 1
•Define Technical Resiliency Architecture Requirements
Step 2
•Identify Standards/Policy for Software Resiliency
Step 3
•Develop Request for Proposal (RFP)/Statement of Work (SOW) for Space Systems
Step 4
•Define Software Architecture Resiliency Goals
Step 5
•Determine Software Architecture Resiliency Quality Attributes
Step 6
•Perform Architecture Analysis of Alternatives

- A software resilient architecture contributes to higher assurance functionality
  – *The system is more resilient and can operate correctly in the presence of most faults and vulnerabilities in the software*

Point of Contact:

Dewanne M. Phillips
The Aerospace Corporation
Dewanne.M.Phillips@Aero.Org
571-307-3765
http://www.aerospace.org/

References

- **[1]** Alexander, J., Shokri, E., Tinto, M., Yee, R., Belz, F., & Scruggs, E. (2013). *Cyber Resilience Assessment Framework.* El Segundo: The Aerospace Corporation. http://www.aerospace.org/research/mission-assurance/cyber-security/
- Barbacci, M., Ellison, R., Lattanze, A., Stafford, J., Weinstock, C., & Wood, W. (2003). *Quality Attribute Workshops (QAWs), Third Edition.* Pittsburgh: Carnegie Mellon University/SEI. http://www.sei.cmu.edu/reports/03tr016.pdf
- Bodeau, D., & Graubart, R. (2011). *Cyber Resiliency Engineering Framework.* MITRE. http://www.mitre.org/publications/technical-papers/cyber-resiliency-engineering-framework
- Brown, M., Kirby, D., Martin, B., & Paller, M. (2011, September 13). *2011 CWE/SANS Top 25 Most Dangerous Software Errors.* MITRE. Retrieved from Common Weakness Enumeration: http://cwe.mitre.org/top25/archive/2011/2011_cwe_sans_top25.pdf
- Carter, A. (2015). *The Department of Defense Cyber Strategy.* Washington DC: DoD. http://www.defense.gov/Portals/1/features/2015/0415_cyber-strategy/Final_2015_DoD_CYBER_STRATEGY_for_web.pdf
- Coley, S., Glenn, R., Kenderdine, J., & Mazella, J. (2014). *Common Weakness Enumeration, CWE Version 2.8.* MITRE. https://cwe.mitre.org/data/
- DNI. (2008). *Intelligence Community Directive (ICD) 503.* Washington: Office of Director of National Intelligence. http://www.dni.gov/files/documents/ICD/ICD_503.pdf

References (Cont.)

- DoD. (2015). Department of Defense Instruction 5000.02. Washington: USD AT&L. http://www.dtic.mil/whs/directives/corres/pdf/500002p.pdf
- DoD. (2003). Department of Defense Instruction, 8500.2; Information Assurance (IA) Implementation. Washington: Department of Defense. http://www.cac.mil/docs/DoDD-8500.2.pdf
- DoD. (2015). FACT SHEET: Resilience of Space Capabilities. Washington: National Security Space Strategy. http://archive.defense.gov/home/features/2011/0111_nsss/docs/DoD%20Fact%20Sheet%20-%20Resilience.pdf
- Dukes, C. (2015). Committee on National Security Systems (CNSS) Glossary, CNSSI No. 4009. Ft Meade: DoD. https://www.cnss.gov/CNSS/openDoc.cfm?Nv/te4i/yL6LBsFHPzMrAw==
- Gates, R., & Clapper, J. (2011). National Security Space Strategy. Washington DC: Department of Defense/Office of the Director of National Intelligence. http://www.dni.gov/files/documents/Newsroom/Reports%20and%20Pubs/2011_nationalsecurityspacestrategy.pdf
- Gen William L. Shelton, U. (2013). Military Space At a Strategic Crossroad. Air & Space Power Journal, 4-10. http://www.au.af.mil/au/afri/aspj/digital/pdf/articles/2013-Sep-Oct/SLP-Shelton.pdf
- Kazman, R. M. (2000). ATAM: Method for Architecture Evaluation. Hanscom AFB: U.S. Department of Defense/Software Engineering Institute. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=5177

References (Cont.)

- Loverro, D. (2015). Space Domain Mission Assurance: A Resilience Taxonomy. Washington: Office of the Assistant Secretary of Defense for Homeland Defense & Global Security. http://fas.org/man/eprint/resilience.pdf
- Pawlikowski, E., Loverro, D., & Cristler, T. (2012). Space: Disruptive Challenges, New Opportunities, and New Strategies. Strategic Studies Quarterly. http://www.au.af.mil/au/ssq/2012/spring/pawlikowski.pdf
- Seacord, R. (2013). Secure Coding in C and C++, 2nd Edition. Addison-Wesley. http://www.cert.org/secure-coding/publications/books/secure-coding-c-c-second-edition.cfm?
- Sklar, J. (2003). Interference Mitigation Approaches for the Global Positioning System. LINCOLN LABORATORY JOURNAL, 14(2), 167-180. https://www.ll.mit.edu/publications/journal/pdf/vol14_no2/14_2interferencemitigation.pdf
- **[2]** Unell, A. (2013). Evaluating Software Architectures for National Security Space Systems. Crosslink, 48-53. https://www.aerospace.org/wp-content/uploads/crosslink/V14N1.pdf
- USGOVT. (2015). National Vulnerability Database. (Department of Homeland Security's National Cyber Security Division) Retrieved from https://nvd.nist.gov/
- Warner, J., & Johnston, R. (n.d.). GPS Spoofing Countermeasures. Los Alamos: Los Alamos National Laboratory. http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-03-6163
- Wilgenbusch, R., & Heisig, A. (2013). Command and Control Vulnerabilities to Communications Jamming. Joint Forces Quarterly, National Defense University, 2nd quarter(69), 56-63. http://ndupress.ndu.edu/Portals/68/Documents/jfq/jfq-69.pdf