

Air
Land
Sea
Space
Cyberspace
Innovation. In all domains.

Bridging Systems and Software Architecture

James L Lewis
Raytheon Certified Architect
7 May 2009

Copyright © 2009 Raytheon Company. All rights reserved.
Customer Success Is Our Mission is a registered trademark of Raytheon Company.

Agenda

- Background
- Architecture Transition Points
- Bridging Systems & Software Architectures
- Project approach
- Key cost drivers and risks
 - Summary of all cost drivers and risks
 - Insufficient Program Planning
 - Artifacts to support transition
 - Lack of Software Architecture Process
- Recommendations for improving transition

Background

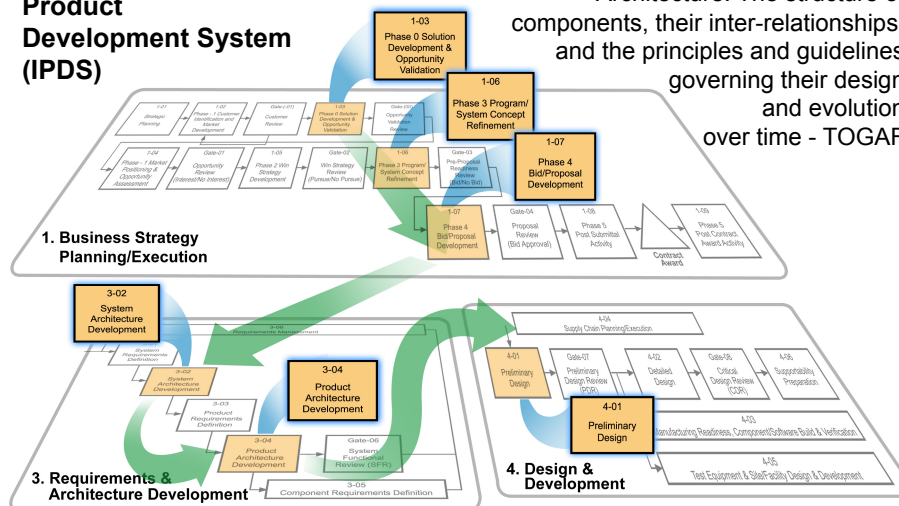
- Raytheon Network Centric Systems (NCS)
 - Continuous Process Improvements in Architecture to Achieve:
 - Reduction in Pursuit Cycle Time
 - Reduction in Engineering Development Cycle Time
 - Increase Engineering Development Productivity
 - Improve Global Workforce Communications
 - Ability to Shape Adaptable Solutions
 - Agility in Handling Change
 - Improvement on NCS Return On Investment Costs (ROIC)
 - Increased Mission Assurance
- Architecture Transitions was identified as a key factor in achieving these goals.

April 6, 2009 | Page 3

Architecture Transition Points

Raytheon Integrated Product Development System (IPDS)

Architecture: The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time - TOGAF



FW09121

April 6, 2009 | Page 4

Bridging Systems & Software Architectures

Raytheon
Network Centric Systems

- Ineffective Architecture transition introduces cost and risk
 - As architecture transition through the various levels of decomposition
 - Late phase rework is needed due to miscommunication of
 - System requirements
 - Architecture principles
 - Quality Attributes desired
- Goal of this Initiative:
 - Identify process improvements
 - Provide guidance to Systems Architects and Software Architects
 - Vertically integrate systems architecture and software architecture products to achieve compliance with
 - System requirements
 - Architecture principles
 - Quality Attributes
 - Reduce cost and risk

April 6, 2009 | Page 5

Project Approach

Raytheon
Network Centric Systems

- Solution Needed To Address Business & Technical Perspectives
- Selected the Zachman Framework.
 - Enterprise = “Capability to transition from Systems Architecture to Software Architecture”
- Primary Mission Architecting Effort
 - mission primitives in a contextual model (Zachman Row 1)
 - business entities in a conceptual model (Zachman Row 2)
 - Association matrices to describe how the business entities relate to each other.
 - list of initiatives were then identified and ranked against the business goals.

	(What) Data	(How) Function	(Where) Location	(Who) People	(When) Time	(Why) Motivation
(Scope/Context) External Influences for Successful Architecture Transition	Business & Eng. Culture, Customer & Program Expectations	OPA, RENP, IPDS		System Stakeholders, e.g. Customer, Managers, COTS/Vendors, Sub-contractors	REAP Events, IPDS Scheduling, REAP Scheduling	NCS Strategy, Program Mgt, Eng Org Goals, Reduce Capture Cycle
(Business Model) The Enterprise – The Successful Architecture Transition	Artifacts involved in Transition	Tailored processes, common language to support transition		System Architecture & Software Architecture	Architecture Transition - From System Arch Dev to PDR/Delta T	Reduce out of phase costs, Increase product quality, reduce schedule risk
(System Model)						
(Technology Model)						
(Detailed Models)						

April 6, 2009 | Page 6

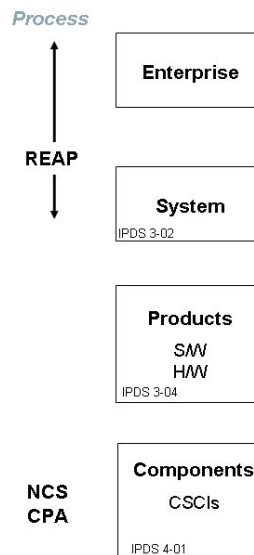
Summary of All Cost Drivers & Risks

- Lack of Software Architecture Process
 - The primary cost driver and risk to the systems to software architecture transition is the deficiency of the software processes involved in the architecture transition and alignment of the processes with systems architecture processes.
- Insufficient Program Planning
 - Program Planning, which includes planning, cost estimation, and scheduling, requires a culture change to ensure the value of architecture is understood and planning frameworks support architecture tasks
- Artifacts to Support Transition
 - The conclusion was that if the work called out in our processes were performed correctly there should not be any appreciable deficiencies. This is not to say that having these documents alone will make for a successful transition.
- Communications during Transition
 - Architecture teams must establish an approach to facilitate the communication, identify the need for a common lexicon, and capture lessons learned from successful (and unsuccessful transitions) to ensure communications are ever improving.
- Governance / Process Enforcement and Awareness
 - This risk addresses how to effectively implement the processes associated with architecture governance and compliance.
- Insufficient Number of Trained Architects for Transition
 - This risk addresses the training necessary for a successful transition. Even though Raytheon has invested in Systems Architecture training curriculum and a Software Architecture training curriculum, there is no training that addresses the transition between Systems Architecture and Software Architecture.

April 6, 2009 | Page 7

Lack of SW Architecture process

- Raytheon Enterprise Architecture Process (REAP) distinctly ends at the system level
 - A Gap between IPDS stages IPDS 3-02 (System) and 3-04 (Product)
- NCS Common Process Architecture (CPA) introduces SW Architecture
 - Integrated with IPDS Stage 4-01 (Component) Preliminary Design
 - Only when MDD is used (i.e., round-trip model to code engineering)
 - Identifies SW Architecture Role, Responsibilities, and Training
- SW Architecture is not integrated across multiple Products or Components
 - Introduces inconsistency, especially when multiple suppliers are involved

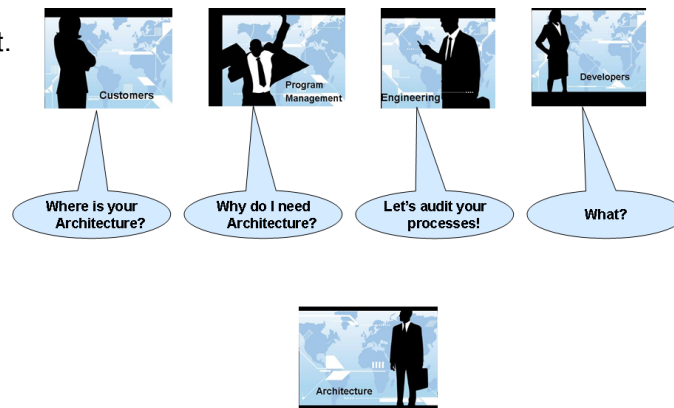


April 6, 2009 | Page 8

Insufficient Program Planning

- Architecture Process has evolved
 - Customer expectation of architecture has evolved
 - Program planning does not always see the value added for architecture
 - Architecture Cost Estimation tools are immature

– As a result.



April 6, 2009 | Page 9

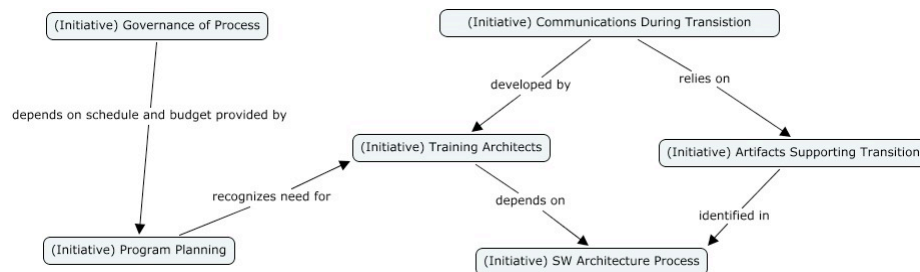
Artifacts to support the transition

- No appreciable deficiencies were identified in artifacts needed for the transition from System to Software Architects
- However –
 - Architecture is a highly communicative process
 - System Architects need to focus the Software Architects on
 - critical information and its significance
 - tribal knowledge that is often difficult to adequately capture in documents
 - reasoning behind trade-offs
 - importance and impacts of the quality attributes

April 6, 2009 | Page 10

Project Recommendations

- Cost drivers and risks are not independent of each other
 - The program planning phase is highly dependent on architects trained on Architecture Planning and cost estimation of architecture.
 - Improved communications during the transition can be achieved by improved training.
 - Improved training is dependent on first establishing software architecture processes.



April 6, 2009 | Page 11

Recommendations To Improve

- Lack of Software Architecture Process
 - Update the NCS CPA Software Architecture Definition
 - Reassign to Product Level Architecture to ensure consistency across Software Products and Components
 - Remove the tie to Model Driven Design

Software Architecture is not only a “model-code” round trip issue
- Artifacts to Support Transition
 - Update REAP to include a “Transition” View
 - TOGAF 8.1.1 includes a Software View that addresses software developer concerns
 - Recommend that this view be built by a combined System and Software Architecture Team.

April 6, 2009 | Page 12

Recommendations To Improve

- Insufficient Program Planning
 - Empower an Architecture Community of Practice
 - Increase communication between those planning the architecture stage and the practicing architects
 - Provide to other programs for processes, tools, and review board conduct
 - Architecture Cost Estimation Tool
 - Program Culture Change

April 6, 2009 | Page 13

Summary

- Architecting is a Pervasive Element of all System Development
- The Architecture must Inform and Govern at all levels of decomposition
- The Organization must improve the process to reduce cost and risk
- The Zachman Framework served us well
 - Characterized the environment of our Enterprise in Row 1
 - Characterized our “Architecting” Enterprise in Row 2
- This project identified several key cost drivers and risks
- Recommendations for improving transition
 - Improved Software Architecture process
 - Create “Transition View” for System to Software
 - Improve Architecture communication between individuals to facilitate better Program Planning

April 6, 2009 | Page 14

Authors

- **James Lewis** (James_L_Lewis@raytheon.com) is a Raytheon Certified Architect, a Senior Principal Systems Engineer at Raytheon, and received his BS - Computer Engineering from the University of Illinois at Urbana-Champaign. James is currently the Chief Architect for the Battle Command and Mission Execution (BCME) package within Future Combat Systems (FCS). In this capacity he works with the FCS LSI to mature the fundamental architectural principals underlying the collaboration of software services across mobile ad-hoc networks constrained by limited quality of service and bandwidth. His 25 years with Raytheon includes leadership of battle command IR&D projects, and engineering associated with communications systems and Air Traffic Control systems.
- **Anne-Marie Buibish** (Anne-Marie_Buibish@raytheon.com) is a Raytheon Certified Architect and Engineering Fellow at Raytheon, and received her BS and MS in Computer Science from Purdue University. With more than 20 years experience in systems and software development in the defense industry, Anne-Marie has a diverse mix of domain experience including Command and Control, F-15 Weapon Control, Tactical Communications and Networking, Modeling and Simulation, and Software and Systems Architecture. Anne-Marie has been a pioneer in the application of emerging technologies and concepts, and has special interest in intelligent systems and application of this technology to policy-based systems. Her current role as Systems Architect combines her diverse experience to provide quality systems for the warfighter.
- **Beth Penisten** (Elizabeth_A_Penisten@raytheon.com) is a Principal Software Engineer at Raytheon and has BS in Mathematics and Computer Science. Beth has over 20 years of experience in Operational and Tactical Battle Management software systems development, including Command and Control, Communications, Software Infrastructures, and Distributed Computing Systems. Beth's Battle Management domain experience spans all phases from operational deliberate planning to observer battle damage assessments. Beth specializes in Software Architecture, Service Oriented Architecture, Modernization of Legacy Systems, and is currently involved in the expansion of Battle Management into new domains.
- **Caleb Conley** (Caleb_E_Conley@raytheon.com) is a Senior Software Engineer at Raytheon and has a BS in Computer Science from Ohio Northern University. He is currently assigned to the Zumwalt Destroyer Program Command and Control Element. Caleb has 8 years of experience in software engineering in Command and Control software.
- **Amy Lange** (Amy_K_Lange@raytheon.com) is a Principal Software Engineer at Raytheon and has BSEET from Purdue University. Amy spent over 12 years designing RF components for transmitters and receivers for Electronic Warfare systems. She then transitioned to Software where she has spent the last 12 years architecting, designing and coding large software systems. She has experience in all phases of development from maintaining a legacy system, transitioning an IR&D developed system to the field and creating new systems.