

SATURN 2014
Portland, Oregon | May 5–9, 2014

How to Incorporate Software Architecture into your Business Model

Raytheon
Intelligence, Information and Services (IIS)
Tim Kertis

May 2014

Overview

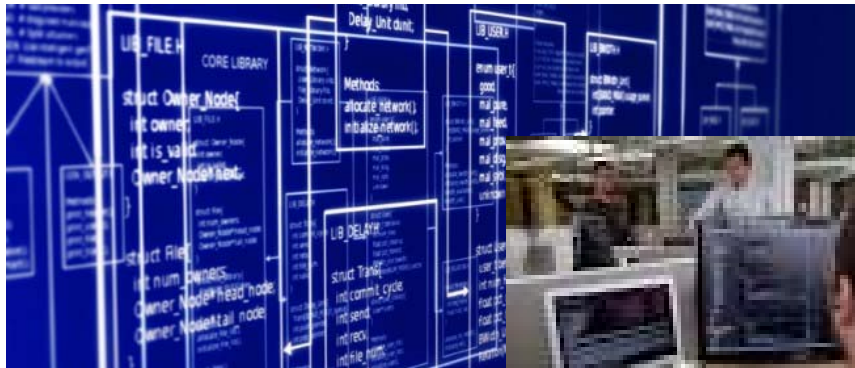
- Background
- The Catalyst for Software Architecture
- Identifying the Need for Software Architecture
- Making the Business Case for Software Architecture
- Establishing the Software Architecture Program
- Lessons Learned
- Summary
- Q&A

Background

- **Who** (am I) ?:
 - Tim Kertis, Principal Software Engineer, Software Architecture and Innovation Department @ Raytheon, Master of Science in Computer and Information Science from Purdue, SEI Software Architecture Professional, 30 years of experience in software development
- **What** (topic am I presenting) ?:
 - The evolving Software Architecture Program @ Raytheon, Intelligence Information and Services (IIS)
- **Where** (does this program apply) ?:
 - Raytheon, IIS, Indianapolis, IN
- **How** (did I get involved in this effort) ?:
 - I initiated interest in a Software Architecture program as a response to the growing code size and complexity of US government and military systems.
 - It all began when I was tasked with the conversion of a simple desktop application into an enterprise-wide global network.
- **Why** (am I presenting) ?:
 - I am presenting this topic to share the experience and lessons learned and further the development of software architecture as a discipline across the industry.

The Catalyst for Software Architecture

- Software Size
- Software Complexity
- Safety, Security and Criticality



Software size and complexity ... is the primary catalyst for the discipline of Software Architecture.

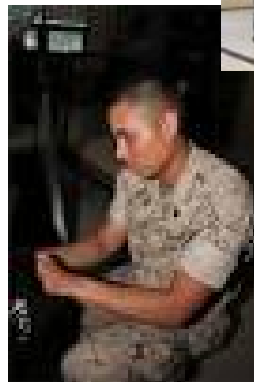
The V-22 AMEGS Story

- V-22 Aircraft Maintenance Software Application
- Transformation from a Standalone System to a Enterprise-Wide Global Network



The Standalone System

- The V-22 Aircraft Maintenance Event Ground Station (AMEGS)
 - Reads a data cartridge taken from the aircraft after flight operations
 - Analyzes and stores the data post flight
 - Produces reports and suggested maintenance activity
 - Limited to processing for one aircraft



The Enterprise-Wide Global Network

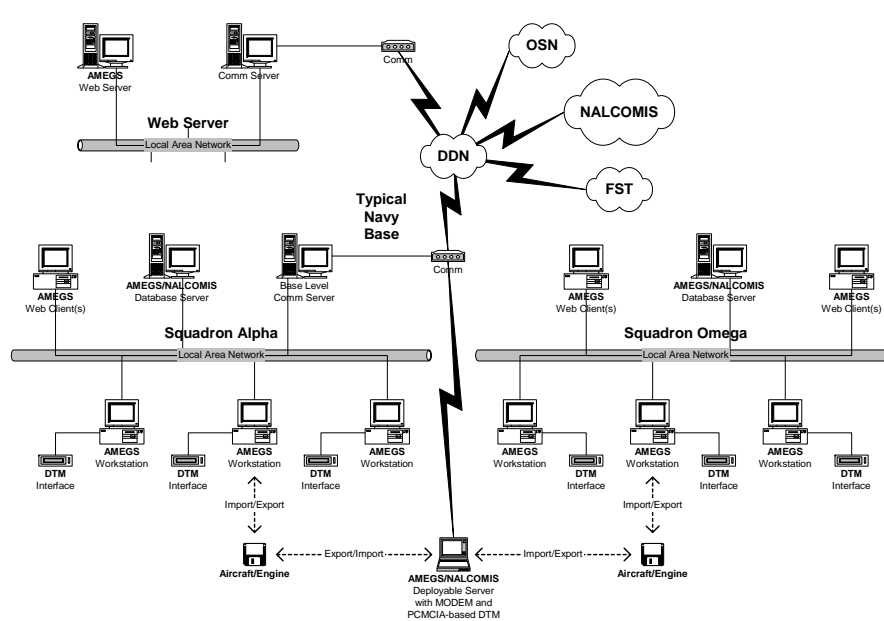
■ AMEGS Web

- Supports data transfer in conjunction with aircraft deployments
- Supports maintenance data analysis across the entire fleet
- Identifies areas of potential for modification to the aircraft
- Predecessor of today's Comprehensive Automated Maintenance Environment-Optimized (CAMEO)

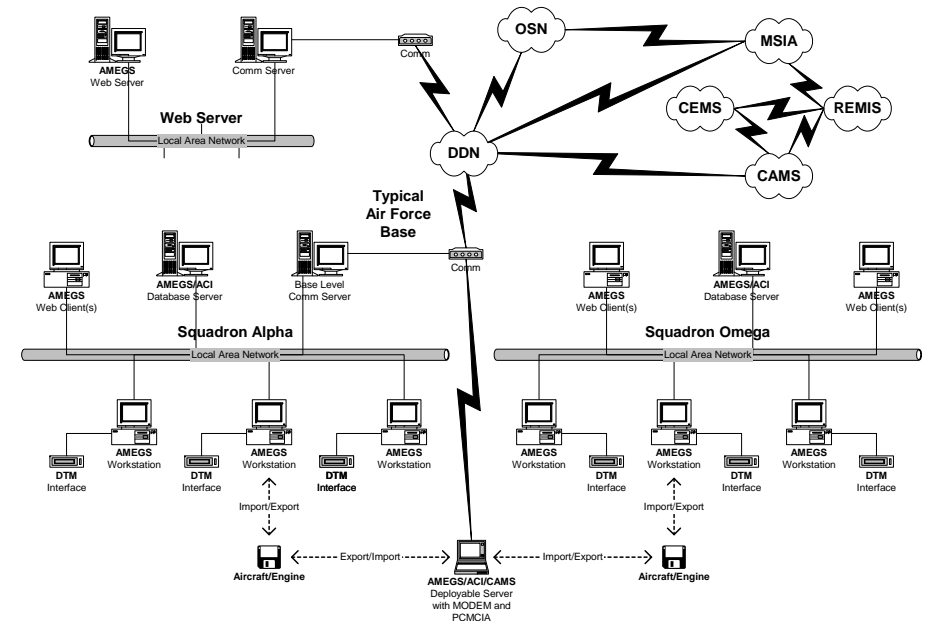


AMEGS Web

■ Complexities in Customer-Specific Network Configurations



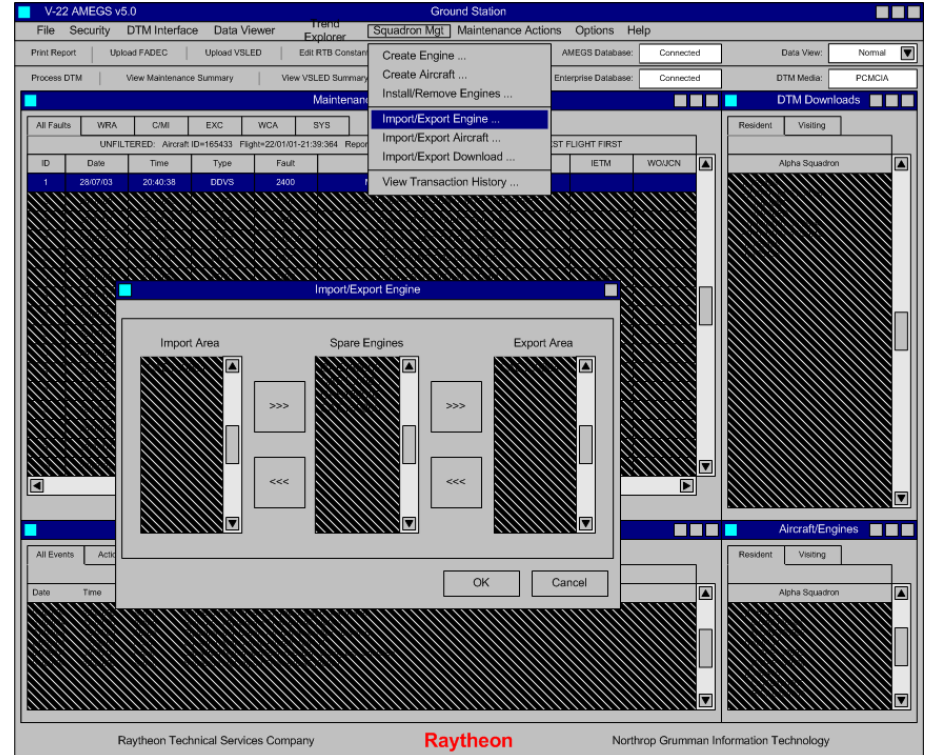
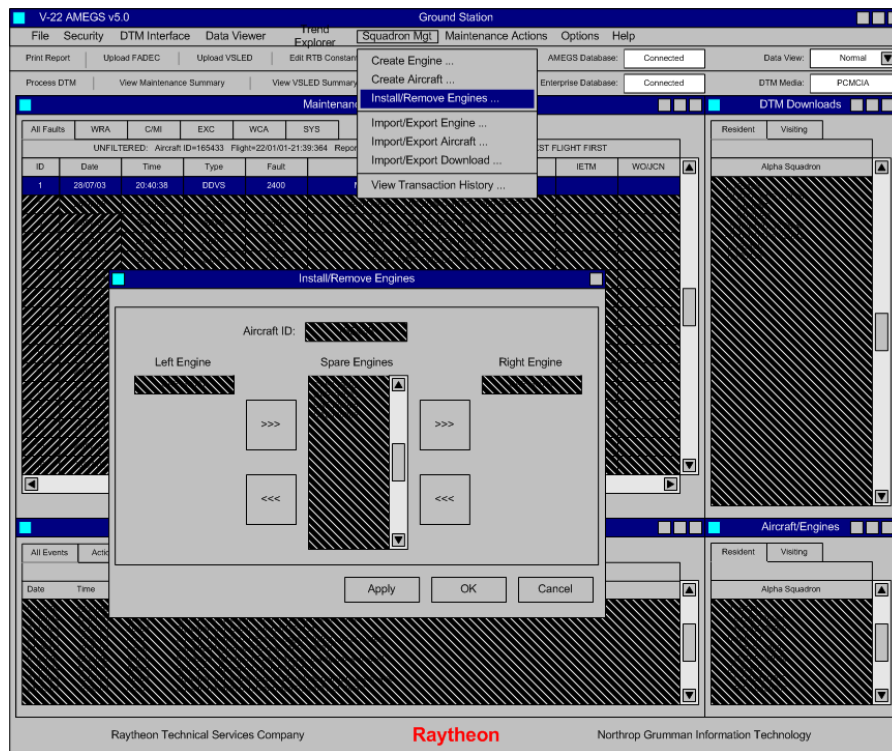
MV-22 U.S. Marines
Aircraft Maintenance Data System Architecture



CV-22 U.S. Air Force
Aircraft Maintenance Data System Architecture

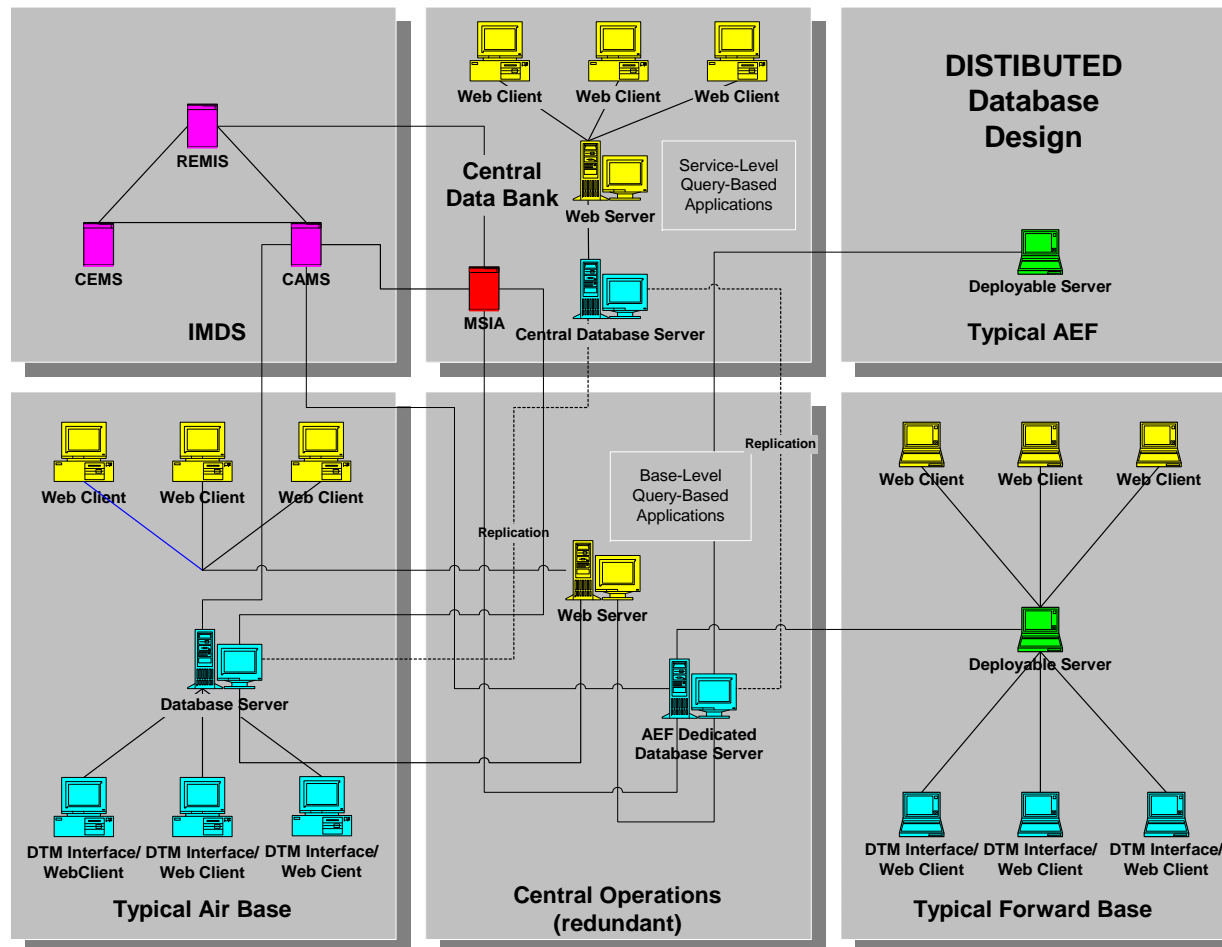
AMEGS Web

■ Complexities in the User Interface



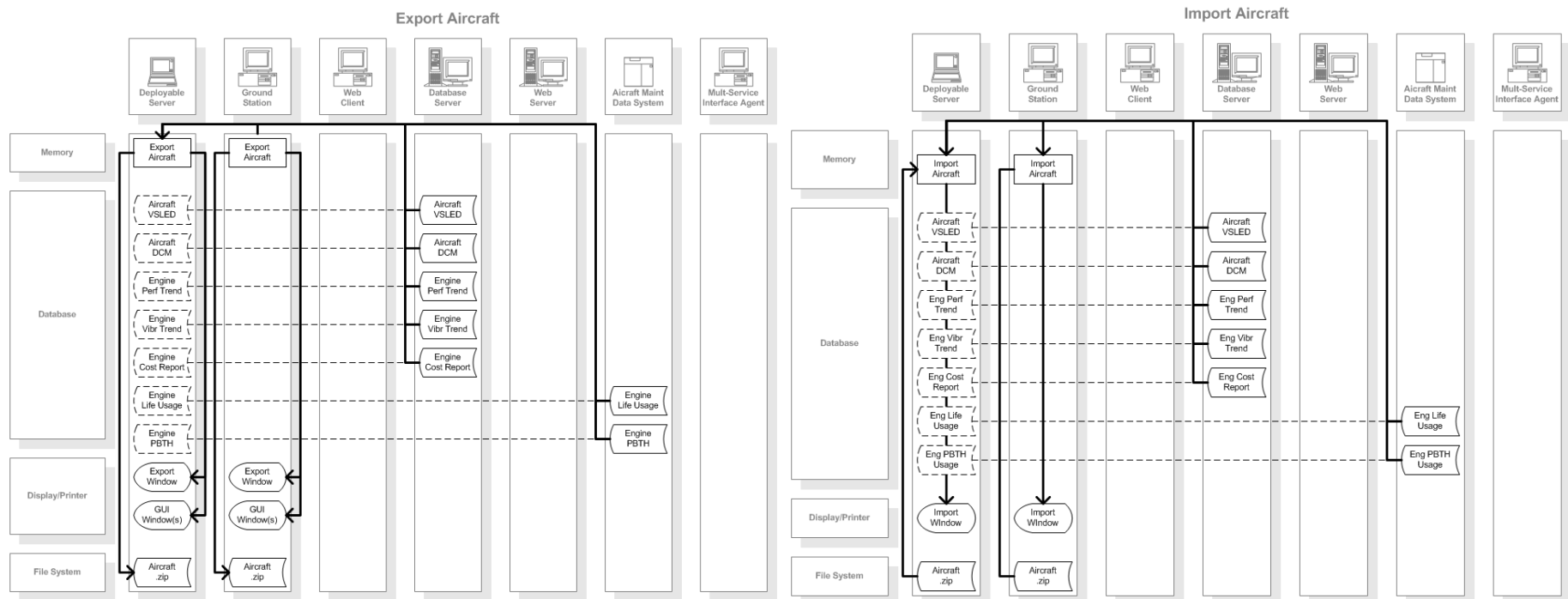
AMEGS Web

- Complexities in Computing Platforms and Network Connectivity



AMEGS Web

■ Complexities in Data Management



Identifying the Need for Software Architecture

- **Code Size and Complexity sources ...**
 - Real-Time Embedded Development/Hard Deadlines
 - Concurrency/Multiple Threads/Performance Requirements
 - Parallel Computing/Multi-Core Processing
 - Multiple Processes/Inter-Process Communication
 - COTS/GOTS Integration/Open Source Development
 - Relational Database Technology/Distributed Databases
 - Multiple Programming Languages/Bindings
 - Multi-Platform Development/Disparate Operating Systems
 - Software Product Lines /System of Systems
 - Software Safety and Security Requirements

**Identify *your* products' sources of
software size and complexity**

Making the Business Case for Software Architecture

- **Present it to:**
 - Engineering
 - Business Development
 - Executive Management
- **Illustrate the Benefits of a:**
 - Software Architect Program
 - Software Product Lines
- **Prepare a Cost/Benefit Analysis**
- **Prepare Cost and Schedule for a Phased Implementation**
- **Obtain Management Buy-In and Budget**

Present a solid business case to *your* organization to justify the investment in Software Architecture.

Establishing the Software Architecture Program

- Products
- People
- Processes

Create a program that promotes quality *products* with capable *people* using repeatable *processes*.

Software Architecture Program

- Software Product Support
- Software Product Lines
- Software Architecture Review Board
- Software Architect Roles and Responsibilities
- Software Architect Competency Model
- Software Architect Certification Program
- Software Architect History and Evidence Package
- Software Architect Certification Board
- Software Architecture Processes & Procedures

Identify the critical elements of **your**
software architecture program.

Identify Software Products

- Identify your systems and software products.
- You will need this information to construct a competency model for your architects.
- You will also need this information for establishing common architectures in support of product lines.

Start by identifying and describing *your* system and software products.

Establish Software Product Lines

- Examine the complete list and description of all of your system and software products.
- Identify where you would benefit from the development of common software architecture to support your product lines.
- Develop a cost/benefit or ROI analysis to justify the investment in a common software architecture.
- Pitch the idea to engineering, business development and executive management.
- Obtain funding to pursue the development of a common software architecture to support your product lines.
- Implement software architecture for your product lines and place it under configuration management.

**Leverage the use of software product lines
where it makes sense for *your* business.**

Establish a Software Architecture Review Board

- Establish a board of software architects to formally review software architecture designs and related artifacts.
- Select software architects from an available pool of certified software architects.
- Review significant software development efforts in the software architecture phase.
- Review the product line development efforts. This board will have authority for changes to the product line architecture.
- Document the review process.

Formally review **your** software architectures.

Define Software Architecture Roles and Responsibilities

- Software Architect Candidates
- Certified Software Architects
- Chief Software Architect
- Software Architect Certification Board Members
- Software Architecture Review Board Members

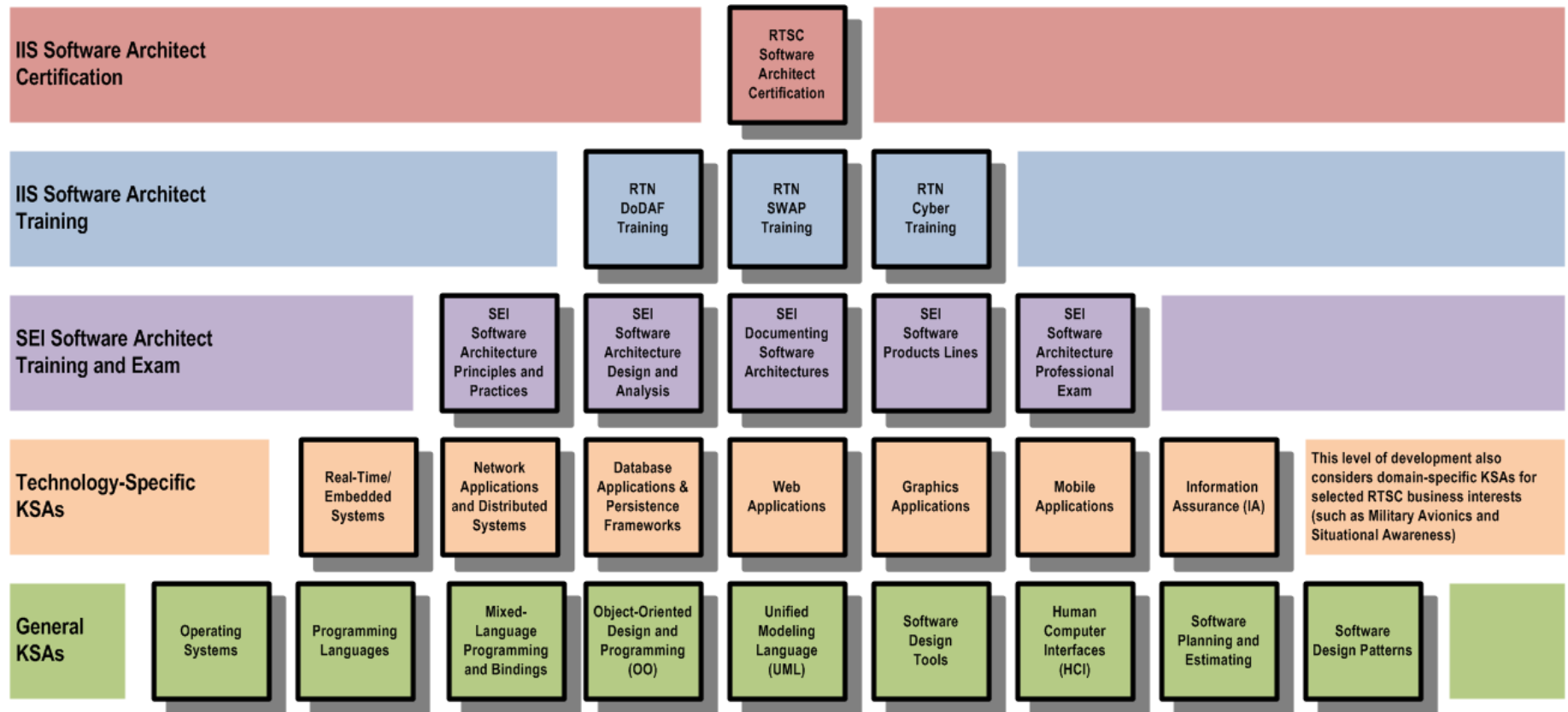
Define roles and responsibilities
for **your** architects.

Design and Document a Software Architect Competency Model

- Education
- Knowledge, Skills and Experience (KSAs)
- Training
- Testing
- Board Certification

Create the competency necessary to architect
your software products.

Software Architect Competency Model (proposed)



Establish a Software Architect Certification Program

- Design and document a software architect certification program.
- Support the program via a competency model that is fine-tuned to your products.
- Document the program and place it under configuration management.
- Make the program a formal process.
- This board will have authority over changes to the software architect certification program.

Certify the competency of *your* architects.

Prototype a Software Architecting History and Evidence Package

- Create a template for the history and evidence package.
- Reuse the template each time a new candidate creates his/her own package.
- Modify the template over time as required.

Document ***your*** architects' competency.
Expedite ***the*** certification process.

Establish a Software Architect Certification Board

- Establish a board of software architects to formally review software architect candidates with their history and evidence packages.
- Initially start with senior software engineers selected by managers and over time transition to certified software architects. Out of necessity, it also may need to include software managers.
- Review software architecture candidates prior to awarding board certification.
- This board will have authority for certifying software architects.
- Document the certification process.

Certify the competency of *your* architects.

Develop and Document Software Architecture Processes & Procedures

- You will need to modify your software organization's software processes to accommodate this new software architecture phase.
- You will also need to document the process of certifying software architects.
- Written procedures may be required to detail subcomponents of those processes.
- Document all processes and procedures and place them under configuration management. The software architecture review board will have authority over any changes to those documents.

Document **your** architecture processes.

Software Architecture Procedures

- Inputs
- Entrance Criteria
- Outputs
- Exit Criteria
- Stakeholders

Define entrance/exit criteria for each of *your* software architecture procedures.

Software Architecture High Level Procedures

- Software Quality Attributes
- Key Architectural Decisions
- Architectural Design
- Software Cost Estimating
- Software Architecture Reviews

Identify the high-level procedures of *your*
software architecture process.

Software Quality Attributes

- Identify software quality attributes for your system.
- Prioritize the software quality attributes for your system.

**Define/document the selection of
software quality attributes for **your** systems.**

Key Architectural Decisions

- Hardware/Drivers/OS
- Software Technology
- Development Tools
- Reuse Strategy

Organize **your** key architectural decisions
by these categories.

Key Architectural Decisions

Hardware/Drivers/OS	Software Technology	Development Tools	Reuse Strategy
Select RTOS/BSPs/Hypervisor	Select IPC Mechanisms	Select IPC Development Tools	Select Software Processes
Select Bus Communication Card/Drivers	Select Bus Communication Protocols	Select Bus Communications Development Tools	Select Historical Productivity Cost Estimation Data
Select Database API Drivers	Select Persistence Technology	Select Database/Persistence Frameworks	Select Software Design Patterns
Select Graphics Card/Drivers	Select Graphics Technology	Select Graphics Development Tools	Select External Software Components
Select Operating Systems	Select Programming Languages and Mixed Language Programming	Select Integrated Development Environments	Select Internal Software Components
Select Hardware Platforms	Select Software Design Methodology	Select Software Design Tools	Select Product Line Architecture

Selection of Hardware/Drivers/OS

- RTOS/BSPs/Hypervisor
- Bus Communication Card/Drivers
- Database API Drivers
- Graphics Card/Drivers
- Operating Systems
- Hardware Platforms

Hardware/Drivers/OS Examples

Key Architectural Decisions	Generic Examples	Vendor-Specific Examples
Select Hardware Platform(s)	Pentium 2.7 GHz processor	Intel Pentium Dual-Core E5400 Wolfdale 2.7GHz LGA 775 65W Dual-Core Desktop processor
Select Operating System(s)	Linux	Red Hat® Enterprise Linux™ Server Version 6.0
Select Graphics Card Driver(s)	ATI HD video and display technology	AMD® ATI Radeon™ HD 5970 and accompanying AMD® Catalyst™ driver
Select Database API Driver(s)	Java Database Connectivity (JDBC) driver	Oracle® Database 10g Release 2.0 JDBC™ Driver
Select Bus Communication Card/Drivers	MIL-STD-1553card with a PCI form factor	Data Device Corporation (DDC) AceXtreme® MIL-STD-1553 PCI and cPCI Card BU-67X10i/T
Select RTOS/BSP/Hypervisors (RTOS)	Secure POSIX-certified with ARINC-653 compliant APEX interface	Green Hills® Integrity™ 178-B Level A
Select RTOS/BSP/Hypervisors (BSP)		Wind River® VxWorks 6.0 BSP for Intel Xeon 5500 series
Select RTOS/BSP/Hypervisors (Hypervisor)	Type 1 Hypervisor	LynuxWorks® LynxSecure™ Hypervisor and Separation Kernel

Selection of Software Technology

- Hardware Platform(s)
- Operating System(s)
- Graphics Card Driver(s)
- Database API Driver(s)
- Bus Communication Card/Drivers
- Real-Time Operating Systems (RTO)
- Board Support Packages (BSP)
- Hypervisors

Software Technology Examples

Key Architectural Decisions	Generic Examples	Vendor-Specific Examples
Select Hardware Platform(s)	Pentium 2.7 GHz processor	Intel Pentium Dual-Core E5400 Wolfdale 2.7GHz LGA 775 65W Dual-Core Desktop processor
Select Operating System(s)	Linux	Red Hat® Enterprise Linux™ Server Version 6.0
Select Graphics Card Driver(s)	ATI HD video and display technology	AMD® ATI Radeon™ HD 5970 and accompanying AMD® Catalyst™ driver
Select Database API Driver(s)	Java Database Connectivity (JDBC) driver	Oracle® Database 10g Release 2.0 JDBC™ Driver
Select Bus Communication Card/Drivers	MIL-STD-1553card with a PCI form factor	Data Device Corporation (DDC) AceXtreme® MIL-STD-1553 PCI and cPCI Card BU-67X10i/T
Select RTOS/BSP/Hypervisors (RTOS)	Secure POSIX-certified with ARINC-653 compliant APEX interface	Green Hills® Integrity™ 178-B Level A
Select RTOS/BSP/Hypervisors (BSP)		Wind River® VxWorks 6.0 BSP for Intel Xeon 5500 series
Select RTOS/BSP/Hypervisors (Hypervisor)	Type 1 Hypervisor	LynuxWorks® LynxSecure™ Hypervisor and Separation Kernel

Selection of Development Tools

- Software Development Design Tool(s)
- Integrated Development Environment(s)
- Integrated Development Environment(s)
- Graphics Development Tool(s)
- Database/Persistence Framework(s)
- Bus Communications Development Tool(s)
- IPC Development Tool(s)

Development Tool Examples

Key Architectural Decisions	Generic Examples	Vendor-Specific Examples
Select Software Development Design Tool(s)		IBM® Rational Rhapsody™, Sparx Systems® Enterprise Architect™, Microsoft® Visio™ for UML, Object Modeling Group® Visual Paradigm™ for UML
Select Integrated Development Environment(s)	Open Source IDE	Eclipse IDE, NetBeans IDE
Select Integrated Development Environment(s)	Commercial IDE	Microsoft® Visual Studio™, Oracle® WebLogic™, IBM® WebSphere™
Select Graphics Development Tool(s)	OpenGL	DISTi® GL Studio Embedded Systems™ (GL/ES) toolkit
Select Database/Persistence Framework(s)	Relational Database	Oracle® Database™, Oracle® MySQL™, Microsoft® SQL Server™
Select Database/Persistence Framework(s)	Hierarchical Database	IBM® Information Management System
Select Database/Persistence Framework(s)	Object Database	Versant® Object Database, Versant® db4o
Select Database/Persistence Framework(s)	Persistence Framework	Java Persistence Framework, Hibernate
Select Bus Communications Development Tool(s)	Bit-Level Programming Tools	DDC® (Data Device Corporation) AceXtreme™ Software, Ballard Technology® Copilot™ Software, Excalibur® Merlin™ Software
Select IPC Development Tool(s)	Berkley TCP/IP Sockets API, Unix RPCGen	Community OpenORB, RTI® (Real-Time Innovations) DDS™ (Distributed Data System)

Establish Reuse Strategy

- Product Line Architecture
- Internal Software Components
- External Software Components
 - Free and Open Source (FOSS)
 - Commercial Off-the-Shelf (COTS)
 - Government Off-the-Shelf (GOTS)
- Software Design Patterns
- Historical Productivity Cost Estimation Data
- Software Processes

Constructing Software Architecting History and Evidence Packages

- Develop the requirements for recording history and evidence package that describes education, experience, training, and testing of your software architect candidates applying for certification.
- Create a template to expedite the construction of the certification packages.
- This will make the artifacts consistent.
- They will be easier to assemble and easier to evaluate.
- It will save time and money.

Prepare a template to expedite the construction of history and evidence packages.

Planning, Specifying, Designing and Implementing Software Product Line(s)

- Study your current and future products.
- What things do they have in common?
- Can you gain efficiency by developing a common software architecture to support these products?
- Will that make you more competitive in developing new software applications?
- Sell the idea to management and obtain funding.
- Write good requirements for the software product line.
- Implement and document the product line architecture.
- Apply CM control over all artifacts.

Invest in a Software Product Line where it makes business sense.

Lessons Learned

- Establishing a successful software architect program requires a *champion* from software engineering to push it through all of the obstacles in its path.
- Selling the idea to management requires a great *slide presentation*. Make a convincing *business case*.
- You must obtain *funding* to make the program viable.
- Making incremental progress is OK. This is a sizeable endeavor and will not happen over night. Place artifacts in *CM control* so that you don't lose ground.
- Provide *incentives* for software engineers to become software architects. Also look for candidates that share an interest in software architecture.

Observe just a few lessons from experience.

Summary

- Identify your Need for Software Architecture
- Make your Business Case for Software Architecture
- Establish your Software Architecture Program

Tailor the suggested steps provided here to implement **your** own software architecture program.

Q&A



Stephen_T_Kertis@raytheon.com