Raytheon
Customer Success Is Our Mission

Neglected Aspects of Software Architecture

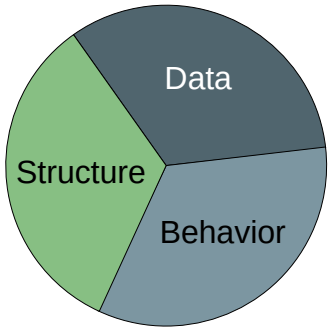
G Todd Kaiser
Chief Architect, Government Systems
Raytheon Intelligence & Information Systems
303-344-6915
gtkaiser@raytheon.com

An Architect's Field of View

Raytheon
Intelligence and Information Systems

- Most architects of software-intensive systems come from the trenches of system development:
  - Software developers
  - Hardware developers
  - System Administrators
  - Database Designers
- Their field of view is usually limited to that which they cultivated in the trenches:
  - System Structure
  - System Behavior
  - System Data

Architect's Technical Field of View



NOTE: Size of "Data" in FOV May Be Exaggerated!

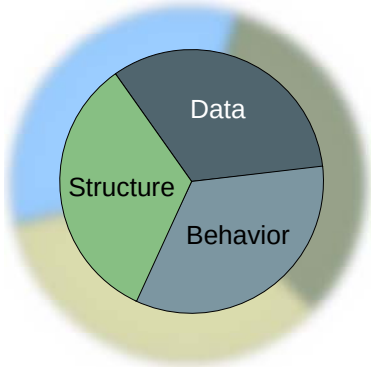
Field of View is Shaped By Our Early Experiences

Limited Field of View

Raytheon
Intelligence and
Information Systems

- Many Architects Focus on Technical Aspects of Architecture Only
- Technical Aspects of Architecture Affect Many Non-Technical Areas
  - Managerial
  - Contractual
  - Financial
- Non-Technical Aspects are Peripheral to the Technical Aspects
- Peripheral Aspects and Technical Aspects Are Symbiotic In Nature

Architect's Field of View Is Often Limited

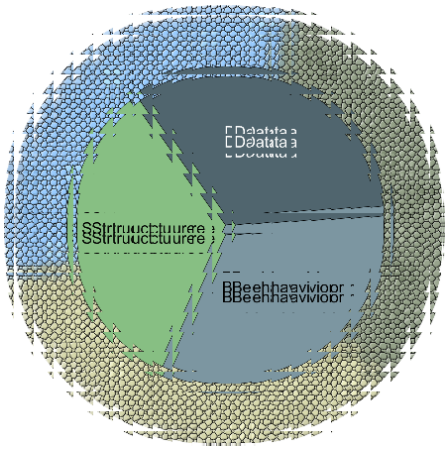


Architects Often Ignore Non-Technical Aspects

"Collateral Damage"

Raytheon
Intelligence and
Information Systems

- Lack of Peripheral Vision Can Lead to "Collateral Damage"
  - Cost overruns
  - Poor Schedule Performance
  - Test & Integration Problems
  - Project Cancellation
  - Litigation
- Damage Caused Can Be So Great That Even the Best Technical Architecture Will Fail
- Damage Usually Manifests As Constraints Imposed by the Technical Aspects of the Architecture

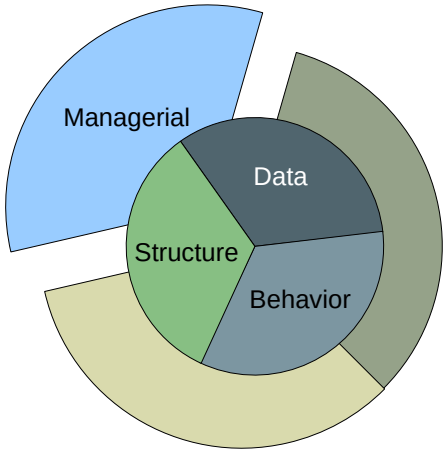


Collateral Damage Can Doom An Architecture

Managerial Aspects

Raytheon
Intelligence and
Information Systems

- Management Provides Vital Enablers:
  - Staff, facilities, communication, etc.
  - Schedules
  - Advocacy
  - Logistics
- These enablers are DIRECTLY affected by architecture
  - Complexity dictated by architecture drives staffing requirements
  - Structure of architecture drives logistics, schedules, and teaming arrangements
  - Breadth of architecture affects communication within development teams

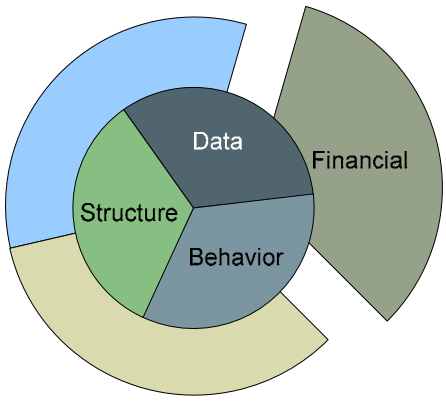


Complex Architectures Can Prove to be Unmanageable

Financial Aspect

Raytheon
Intelligence and
Information Systems

- Development of architectures requires funding
  - Labor, licenses, hardware, facilities
- Highly Coupled Architectures force financial constraints:
  - These architectures usually require lots of funding over short periods of time
- Loosely Coupled Architectures provide financial flexibility
  - The highly modular nature allows development of pieces of the architecture over longer periods of time

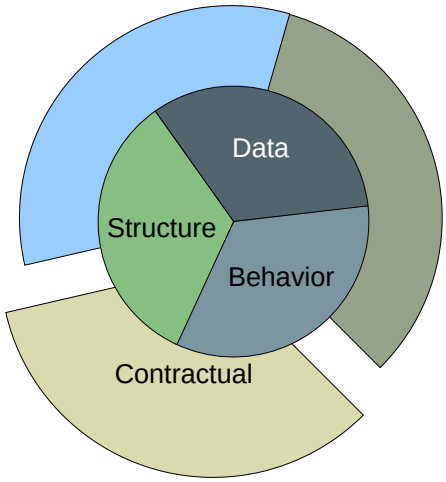


Structure of Architecture Can Handcuff Funding

Contractual Aspect

Raytheon
Intelligence and
Information Systems

- The Structure of an Architecture can affect contracts with:
  - Customers
  - Vendors
  - Teammates
- Architectures with many-to-many dependencies cause complex contract issues:
  - Work is not easily divided amongst teammates and vendors
  - Responsibilities for requirement fulfillment can be unclear
- Acceptance of a highly coupled architecture can be tricky
  - Acceptance might have to wait until all pieces are built – extending work on pieces that could otherwise finish early

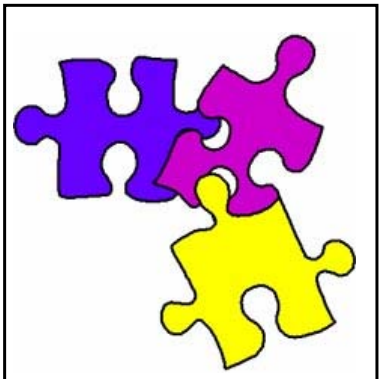


Complicated Structures Cause Complex Contractual Issues

Example #1 – Managerial Aspect

Raytheon
Intelligence and
Information Systems

- CORBA Based Architecture
  - Two architectures were created, both with a great emphasis on the "plug and play" nature of CORBA
  - CORBA IDL would define interfaces for functionality, allowing specific implementations to be "plugged in"
- Problem: CORBA is a complicated technology requiring highly-skilled developers
  - One architecture did not recognize this problem, causing a lack of acceptance of their architecture
  - Other architecture recognized the lack of skilled developers in the organization and minimized the use of CORBA



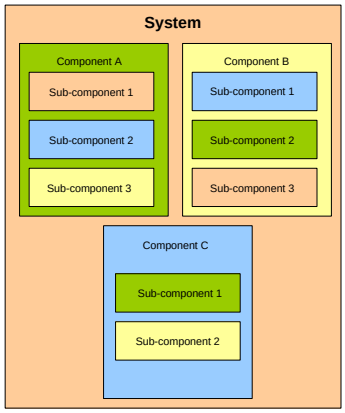
Software Components "Easily" Plug Together

Architectures Must Accommodate Organizational Constraints

Example #2 – Contractual Aspect

Raytheon
Intelligence and
Information Systems

- Non-traditional Structure
  - Architecture was created where small pieces were handed off to separate development organizations
  - Component responsibility was based on each development organization's area of expertise
  - Better organizational alignment was avoided in order to optimize the software architecture
- Problem: Each development organization depended upon each other and was responsible to each other
  - Made contractual relationships very tricky
  - Optimization gained was more than offset by contractual complexity



Colors Indicate Responsibility

Architectures Must Accommodate Work-Share Considerations

Solutions

Raytheon
Intelligence and
Information Systems

- Engage all the stakeholders
  - Managers, contract personnel, schedulers and planners, controllers/business operations personnel
  - Engage these people early and often
  - Learn their concerns and keep them in mind while developing your architectures
- Architecture Development is a Two-Way Street!
  - Don't let these "other" aspects be the sole driver for your architecture!
    - If you need to make a change that might upset the apple cart, do it!
    - Engage these stakeholders, get them on board with the change, help them become part of the solution
    - Don't let things like work-share undermine your architecture
- Architecting is more than just arranging lines and boxes
  - Being a communicator, negotiator, and facilitator are very necessary!

Architecture is More Than Just Technology

Conclusions

Raytheon
Intelligence and
Information Systems

- Architecture is more than just the structure, behavior and data in a system
- Managerial, Contractual, and Financial aspects of an architecture are equally important
- These “other” aspects can make or break your architecture unless you consider the “collateral damage”
- Engage all the stakeholders of your architecture, not just the technology people

Architecture is More Than Just Technology

Biography

Raytheon
Intelligence and
Information Systems

G. Todd Kaiser

Todd Kaiser is the Chief Architect for the Government Systems product line of Raytheon’s Intelligence and Information Systems business. Todd is responsible for the technological aspects of program execution in the area of architecture and software development. The Government Systems business area, based in Aurora, Colorado, builds and maintains ground systems for National Space Systems. In addition, Todd is on staff to the Manager of Software Engineering in Aurora and a member of the Rocky Mountain Engineering Software Council. His research and work on the Software Council extends the knowledge and capability of the software discipline and forms the basis of newly developed architectures, concepts, theories, methodologies and products.

Todd has been working in the National Space Systems arena for the last 12 years for the former Hughes Aircraft Company and now Raytheon. Starting in software development, Todd has developed software in the area of flight dynamics, telemetry and commanding, and mission scheduling. He has also worked in the area of ground systems engineering, developing concepts of operation, requirements, and ground systems architecture. Todd holds a B.S. in physics and mathematics and an M.S. in computer science from the University of Denver.