Architecture-Centric Virtual Integration Practice with AADL

Peter Feiler

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0287

Safety Critical Embedded Software System Challenge

Architecture Centric Virtual Integration Practice with AADL

Embedded Software System Qualification and Assurance

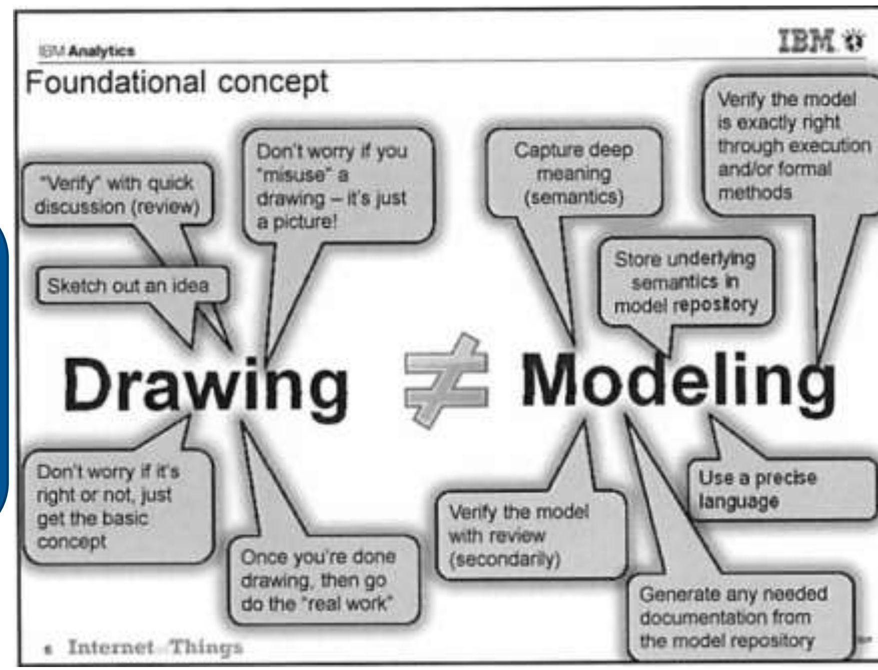
Model Based Engineering (MBE)

Modeling and Simulation have been around for a long time

Continuous and discrete state modeling, verification and code generation of detailed design (Mathworks Simulink, ANSYS SCADE)

Software modeling through UML with limited semantics

Early system modeling with SysML



Our MBE Focus

Safety-Critical Software Systems with stringent Safety, Security, and Performance Requirements

The Safety Critical Embedded Software System Challenge

Problem:

Software increasingly dominates safety and mission critical system development cost.

80% of issues discovered post unit test.

Solution: Early discovery of system level issues through virtual Integration and incremental analytical assurance.

Approach:

International standard based technology matured into practice through pilot projects and industry initiatives.

Open source research prototyping platform continually enhances analysis, verification, and generation capabilities.

Reduced Defect Leakage through Early Analytical Assurance is Critical

We Rely on Software for Safe System Operation

Quantas Airbus A330-300 Forced to make Emergency Landing - 36 Injured

Written by htbw on Oct-7-08 1:48pm

From: soyawannaknow.blogspot.com

Email

Thirty-six passengers and crew were injured, some seriously, in a mid-air drama that forced a Qantas jetliner to make an emergency landing, the Australian carrier and police said on Tuesday.

The terrifying incident saw the Airbus A330-300 issue a mayday call when it suddenly changed altitude during a flight from Singapore to Perth, Qantas said.

Australian Transport Safety Bureau said yesterday. The aircraft dropped 650 feet within seconds, slamming passengers and crew into the cabin ceiling, before the pilots regained control.

``This appears to be a unique event,'' the bureau
Toulouse, France-based Airbus, the world's largest
aircraft, issued a telex late yesterday to airlines th
fitted with the same air-data computer. The adviso
minimizing the risk in the unlikely event of a similar

Embedded software systems introduce a new class of problems not addressed by traditional system safety analysis

FAA says software problem with Boeing 787s could be catastrophic

By **Dan Catchpole**
@dcatchpole

The Federal Aviation Administration says a software problem with Boeing 787 Dreamliners could lead to one of the advanced jetliners losing electrical power in flight, which could lead to loss of control.

The Buzz: Hipster's dilemma
Boeing & aerospace news
Aerospace blog

The FAA notified operators of the airplane Friday that if a 787 is powered continuously for 248 days, the plane will automatically shut down its alternating current (AC) electrical power.

Importance of System and Embedded Software System Co-Engineering

New control system to compensate for bad engineering decision

Two Crashes In Five Months

What's Wrong with Boeing's 737 Max 8?

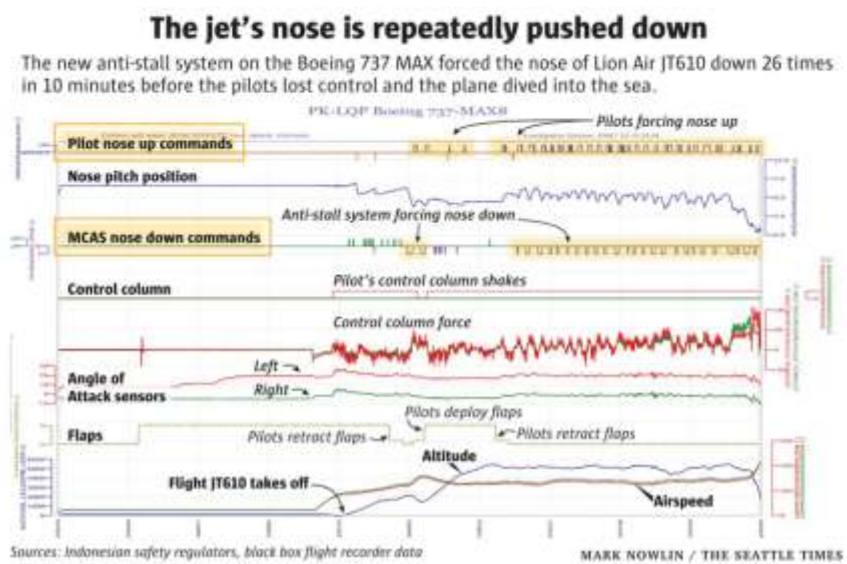
Boeing's new airplane has only been around for two years and already two 737 Max 8s have crashed, killing 346 people. The disasters may be attributable to a design flaw that emerged when engineers began cutting corners.

One of the reasons the Leap engine is so economical is because its air intake has an enormous diameter: 198 centimeters (6.5 feet). While the long-legged Airbus A320neo has plenty of room for such a massive engine, the landing gear on Boeing's Max 8 is short, limiting ground clearance under the wings. The engine simply doesn't fit.

Pressed to come up with a solution, Boeing's star engineers came up with the idea of shortening the engine mount structure, which fastens the heavy engines to the underside of the wings. This did the trick, but it came at the cost of seriously altering the aircraft's flight mechanics. As a result, the Max 8 tended to dangerously raise its nose. Under certain circumstances – rare and extreme, to be sure, yet possible nonetheless – there was a greater chance of the plane stalling and even crashing.

Boeing engineers, in turn, came up with another makeshift solution. They developed a software that would work in the background. As soon as the nose of the aircraft pointed upward too steeply, the system would automatically activate the tailplane and bring the aircraft back to a safe cruising plane. The pilots wouldn't even notice the software's intervention – at least that was the idea. In fact, Boeing didn't even consider it necessary to inform pilots about the newfangled MCAS, or "Maneuvering Characteristics Augmentation System."

MCAS control system fights flight control system and is sensitive to bad sensor input

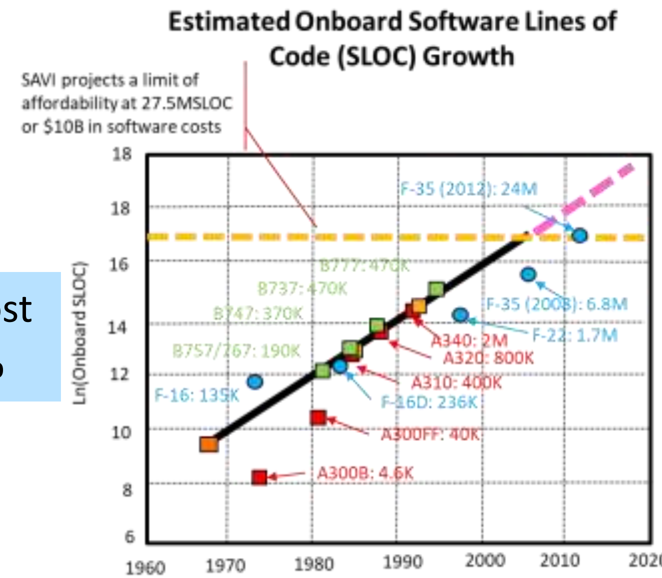


Problem: Growth in Complexity and Late Error Discovery in Cyber Physical Systems is Driving Affordability

Cyber Physical Systems, especially Aviation Systems, are reaching a software affordability limit, impacting the amount of new functionality we can integrate.

Software as % of total system development cost
1997: 45% ➞ 2010: 66% ➞ 2024: 88%

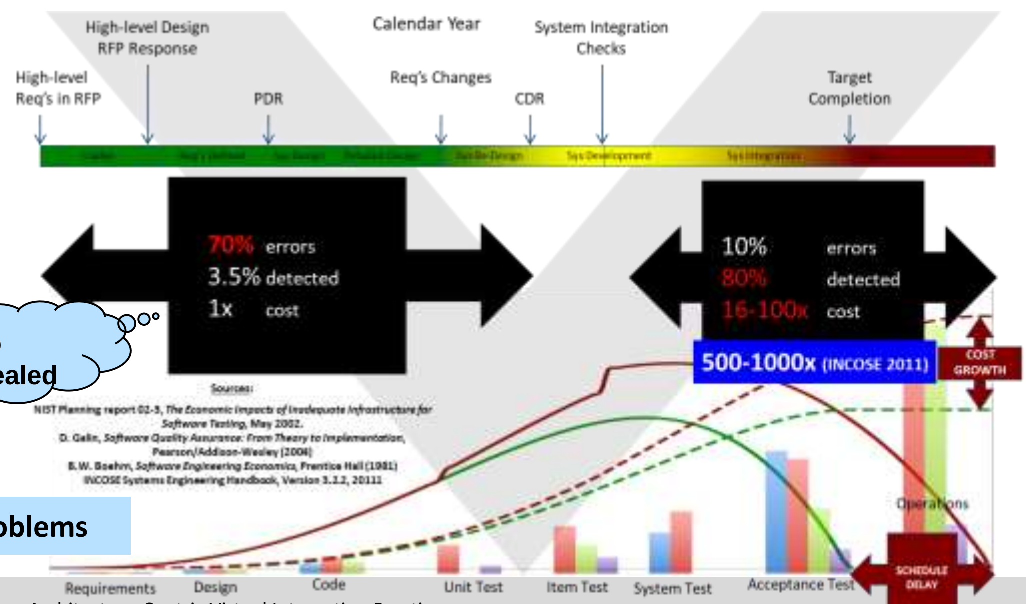
Post unit test software rework currently ~50% of total system development cost



SAVI
System Architecture Virtual Integration
A Commercial Aviation Industry Consortium

This represents a significant opportunity for cost reduction and functional enhancement by discovering issues early through virtual integration and analysis of embedded software system models and synthesis of implementation from verified models.

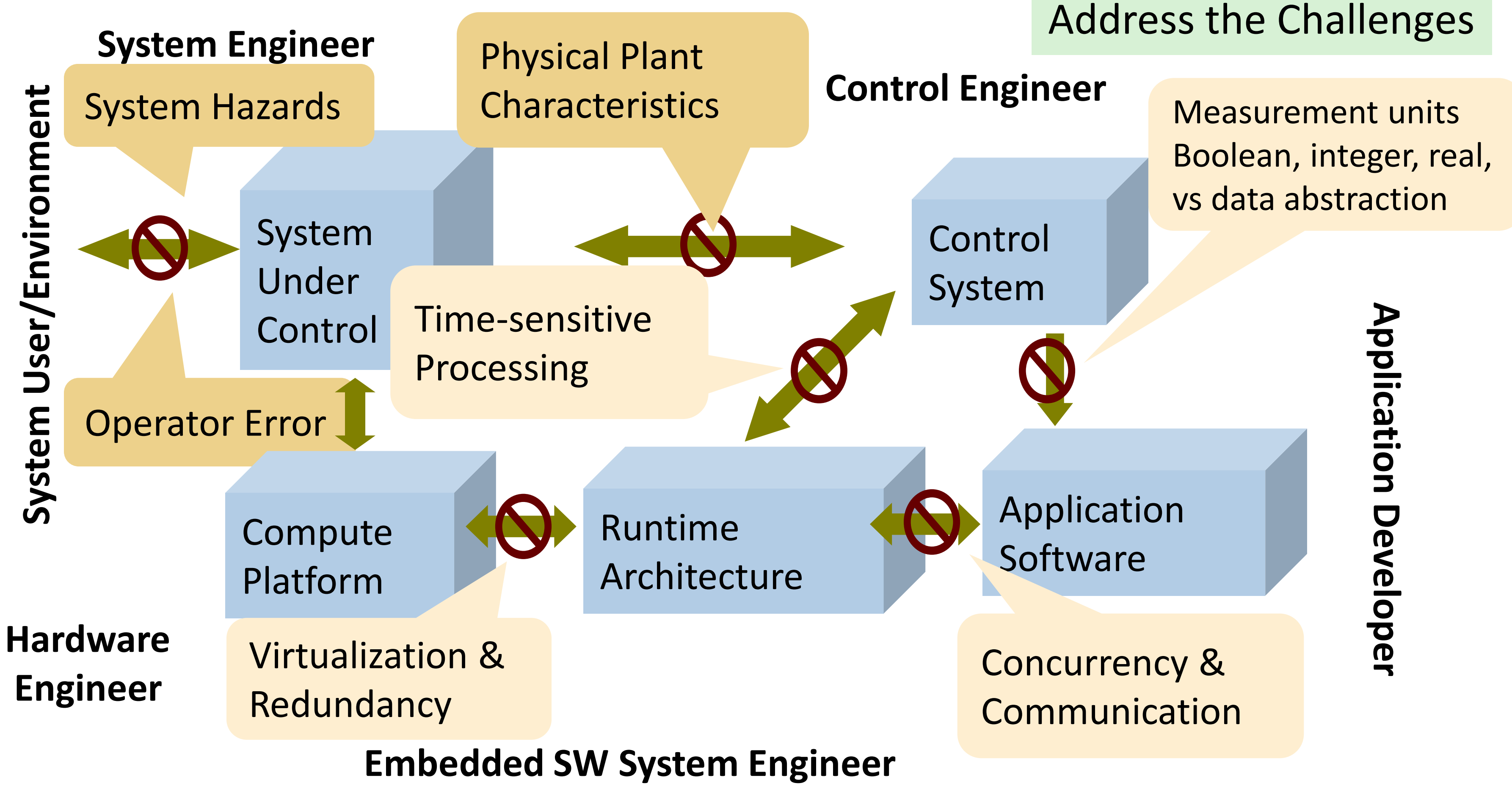
Gap revealed



F35 generated software had integration problems

Technical Challenges in Safety-Critical Embedded Software Systems

AADL Semantics
Address the Challenges



Why do system level failures still occur despite best safety practices?

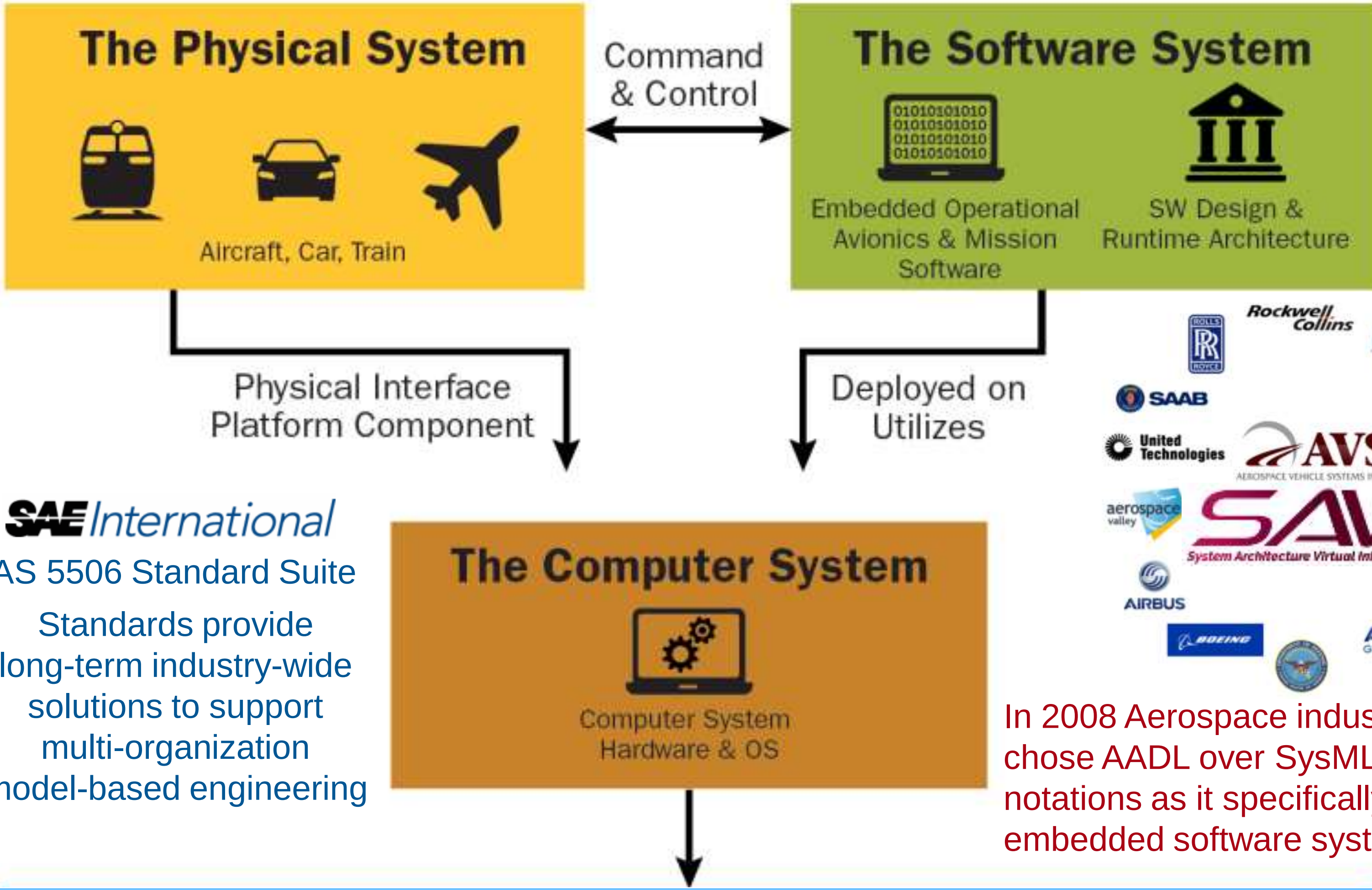
*Embedded software systems have become a major **safety** and **cyber security** risk*

Safety Critical Embedded Software System Challenge

Architecture Centric Virtual Integration Practice with AADL

Embedded Software System Qualification and Assurance

Architecture Analysis & Design Language (AADL) Standard Targets Embedded Software Systems



SAE International

AS 5506 Standard Suite

Standards provide long-term industry-wide solutions to support multi-organization model-based engineering

In 2008 Aerospace industry initiative chose AADL over SysML and other notations as it specifically addresses embedded software systems

AADL captures mission and safety critical embedded software system architectures in virtually integrated analyzable models to discover system level problems early and construct implementations from verified models

SAE International AADL Standard Suite (AS-5506 series)

Core AADL language standard [V1 2004, V2 2012, V2.2 2017]

- Focused on embedded software system modeling, analysis, and generation
- Strongly typed language with well-defined semantics for execution of threads, processes on partitions and processor, sampled/queued communication, modes, end to end flows
- Textual and graphical notation
- Revision V3 in progress: interface composition, system configuration, binding, type system unification

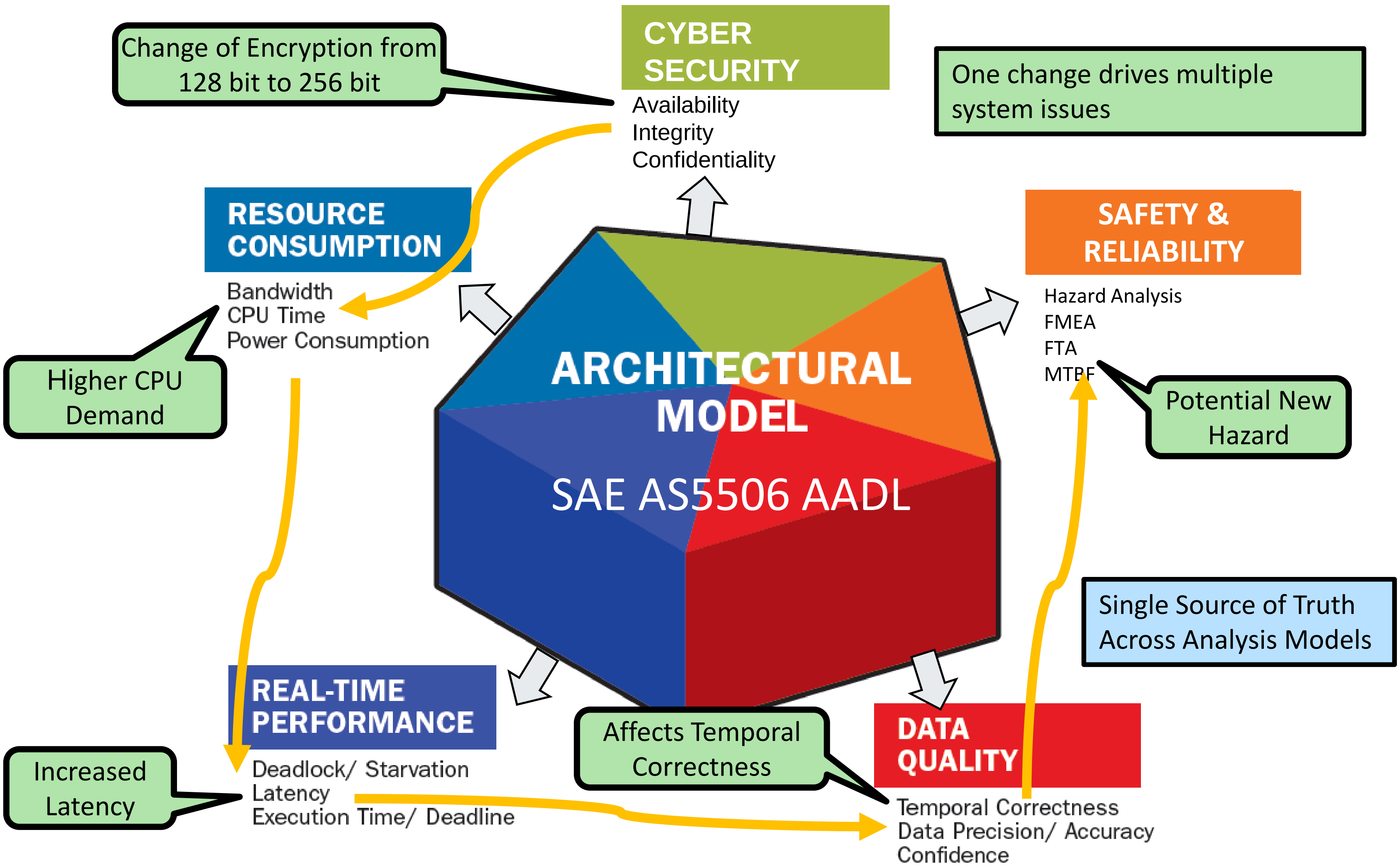
Standardized AADL Annex Extensions

- Error Model language for safety, reliability, security analysis [2006, 2015]
- ARINC653 extension for partitioned architectures [2011, 2015]
- Behavior Specification Language for modes and interaction behavior [2011, 2017]
- Data Modeling extension for interfacing with data models (UML, ASN.1, …) [2011]
- AADL Runtime System & Code Generation [2006, 2015]

AADL Annexes in Progress

- Network Specification Annex
- Cyber Security Annex
- FACE Annex
- Requirements Definition and Assurance Annex
- Synchronous System Specification Annex

Analysis of System Properties via Architecture Model
A Contribution to Single Source of Truth



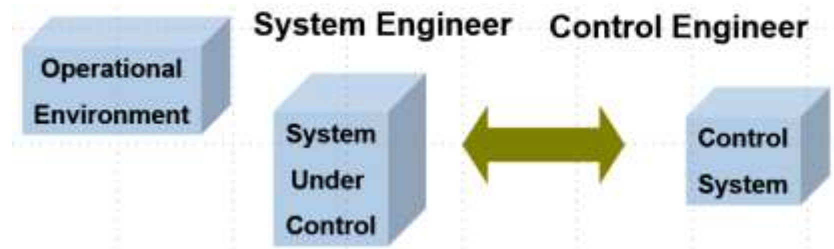
Latency and Jitter Contributors

Control System Engineering View

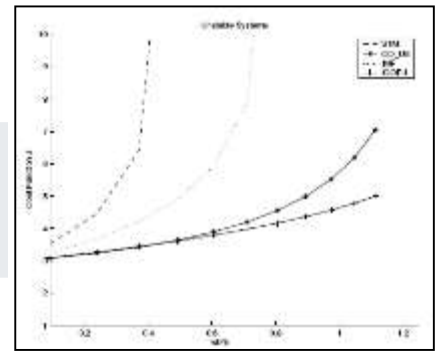
Processing latency

Sampling latency

Physical signal latency



Impact of Scheduler Choice on Controller Stability
A. Cervin, Lund U.
CCACSD 2006



Software System Latency Contributors

Execution time variation: algorithm, use of cache

Processor speed

Resource contention

Preemption

Legacy & shared variable communication

Rate group optimization

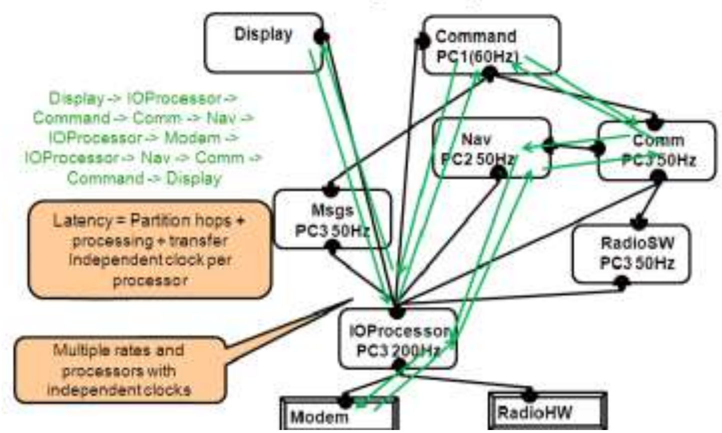
Protocol specific communication delay

Partitioned architecture

Migration of functionality

Fault tolerance mechanisms

Flow Use Scenario through Subsystem Architecture



Demonstrations of Effectiveness in use of ACVIP with AADL

Finding Problems Early (AMRDEC/SEI)

- Summary: 6 Week Virtual Integration on CH47 using AADL
- Result: Identified 20 major integration issues early
- Benefit: Avoided 12-month delay on 24-month program

CH47 Chinook

Commercial Aircraft Industry Addresses Embedded Software System Challenge

SAVI
System Architecture Virtual Integration

- Summary: Industry Consortium Matures Virtual System Integration
- Result**:** Proof of virtual integration concept in 2008-09 led to ten year maturation commitment
- Benefit: ROI study with $2B savings on $10B aircraft through 33% early detection

Transforming procurement (Joint Multi-Role)

- Summary: Industry/DoD mission system architecture demonstrations using ACVIP
- Result: Pre-integration fault identification
- Benefit: 10X reduction integration test cost

Improving System Security (DARPA / AFRL)

- AADL applied to Unmanned Aerial Vehicles & Autonomous Truck
- Result: AADL models enforced security policies and were used to auto build the system
- Benefit: Combined with formal methods verification, prevented security intrusion by a red team

DARPA

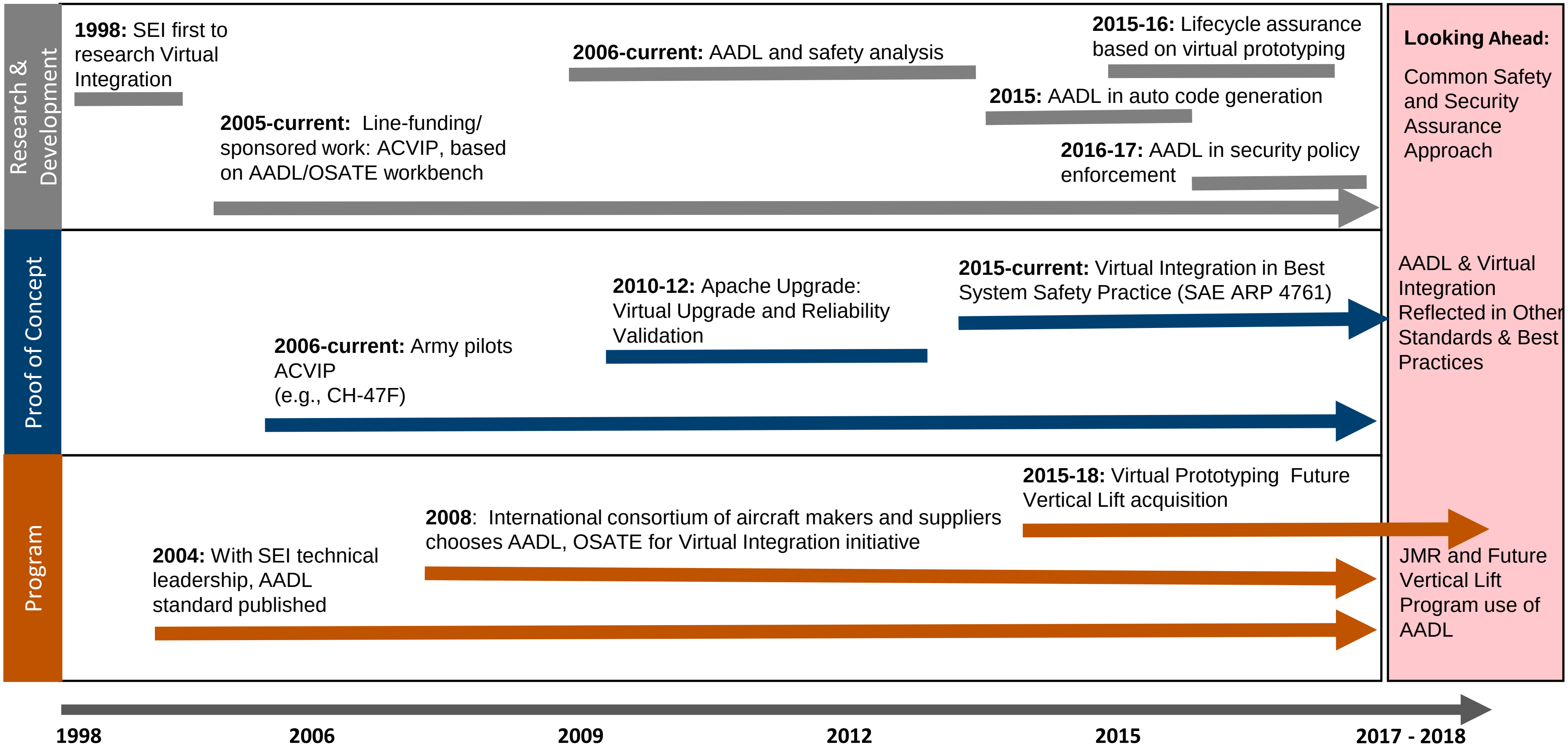
High Assurance Cyber Military Systems (HACMS)

Unmanned Quadcopter

TARDEC Autonomous Truck

Unmanned Little Bird

Virtual Integration & Assurance of Safety-Critical Systems



AADL= Architecture Analysis & Design Language; OSATE = Open Source AADL Tool Environment; JMR = Joint Multi-Role
ACVIP = Architecture Centric Virtual Integration Practice

Safety Critical Embedded Software System Challenge

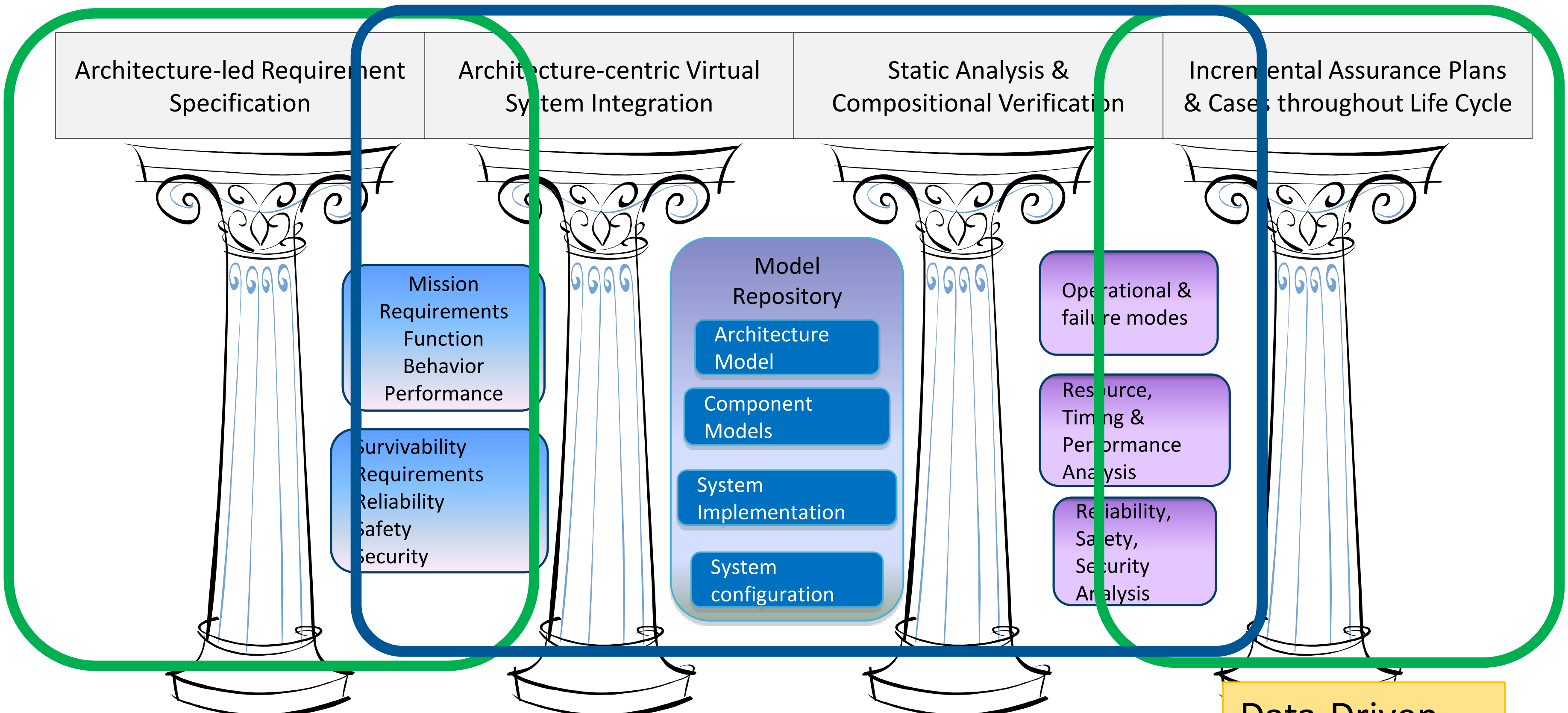
Architecture Centric Virtual Integration Practice with AADL

Embedded Software System Qualification and Assurance

Assurance & Qualification Improvement Strategy

Assurance: Sufficient evidence that a system implementation meets system requirements

2010 SEI Study for AMRDEC Aviation Engineering Directorate



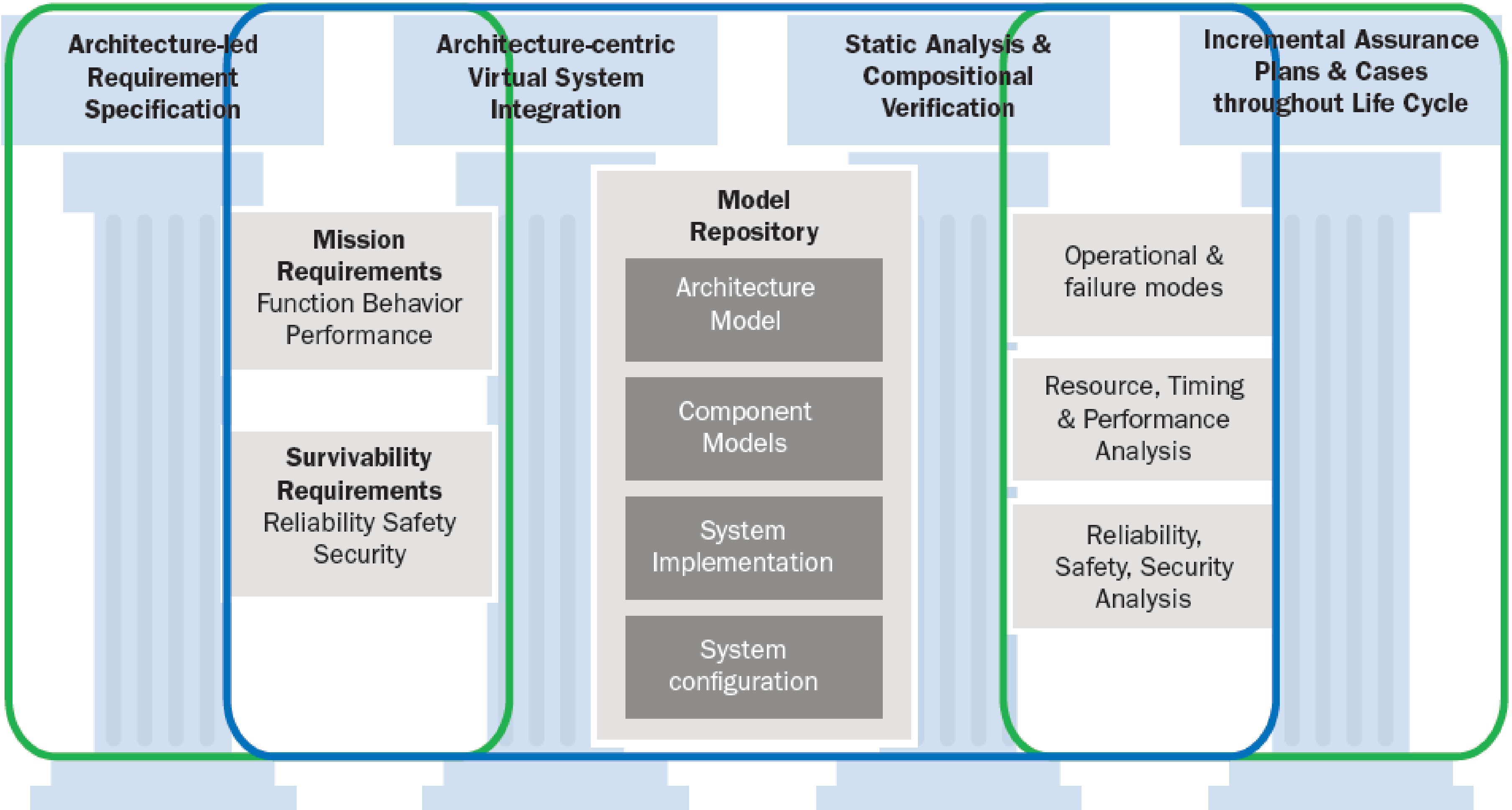
Architecture Centric Virtual System Integration Practice (ACVIP)

Architecture Led Incremental System Assurance (ALISA)

Data-Driven
High Leverage
Cost Effective

Assurance & Qualification Improvement Strategy

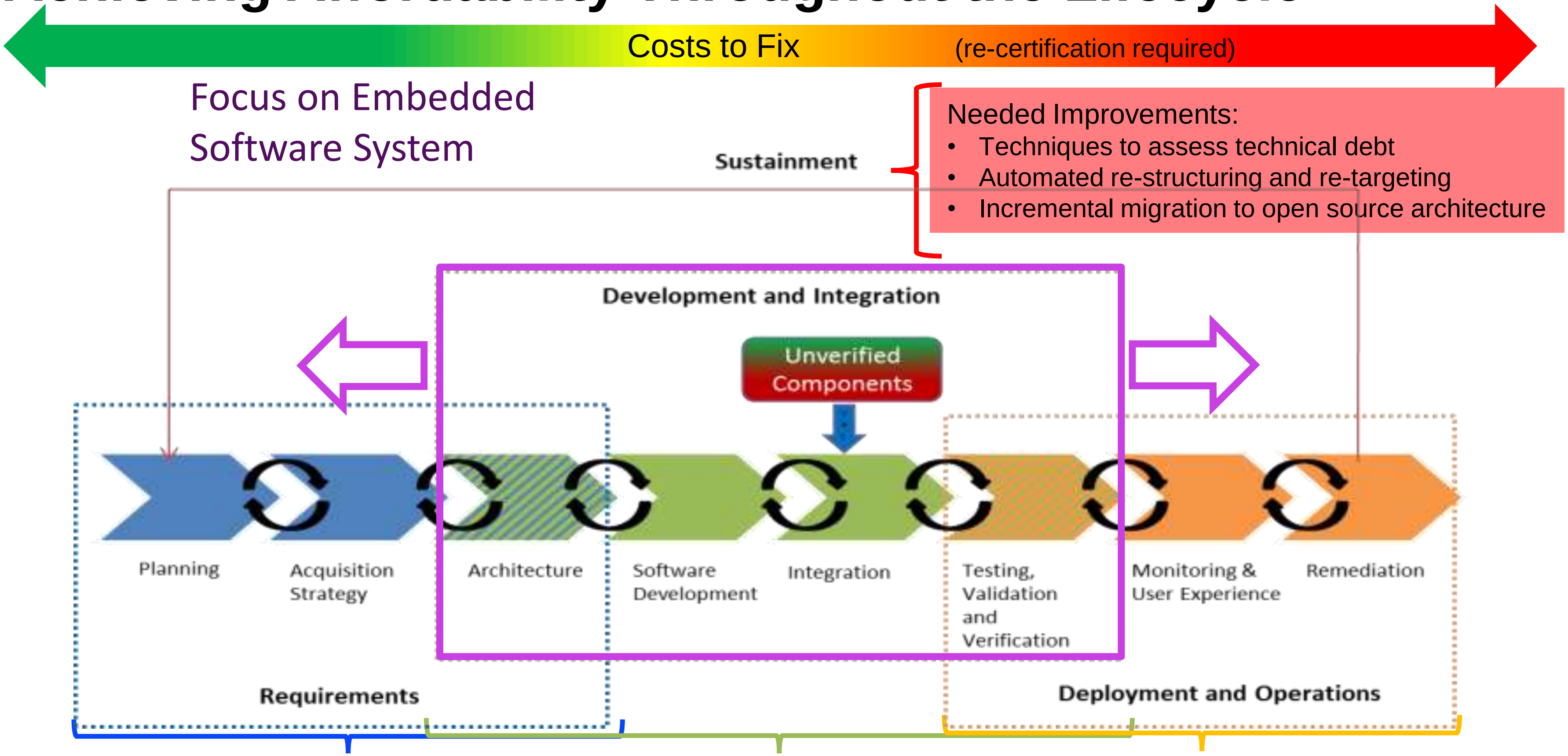
Assurance: Sufficient evidence that a system implementation meets system requirements



Architecture-centric Virtual System Integration (ACVIP)
Incremental Lifecycle Assurance (ALISA)

2010 SEI Study for AMRDEC Aviation Engineering Directorate

Achieving Affordability Throughout the Lifecycle



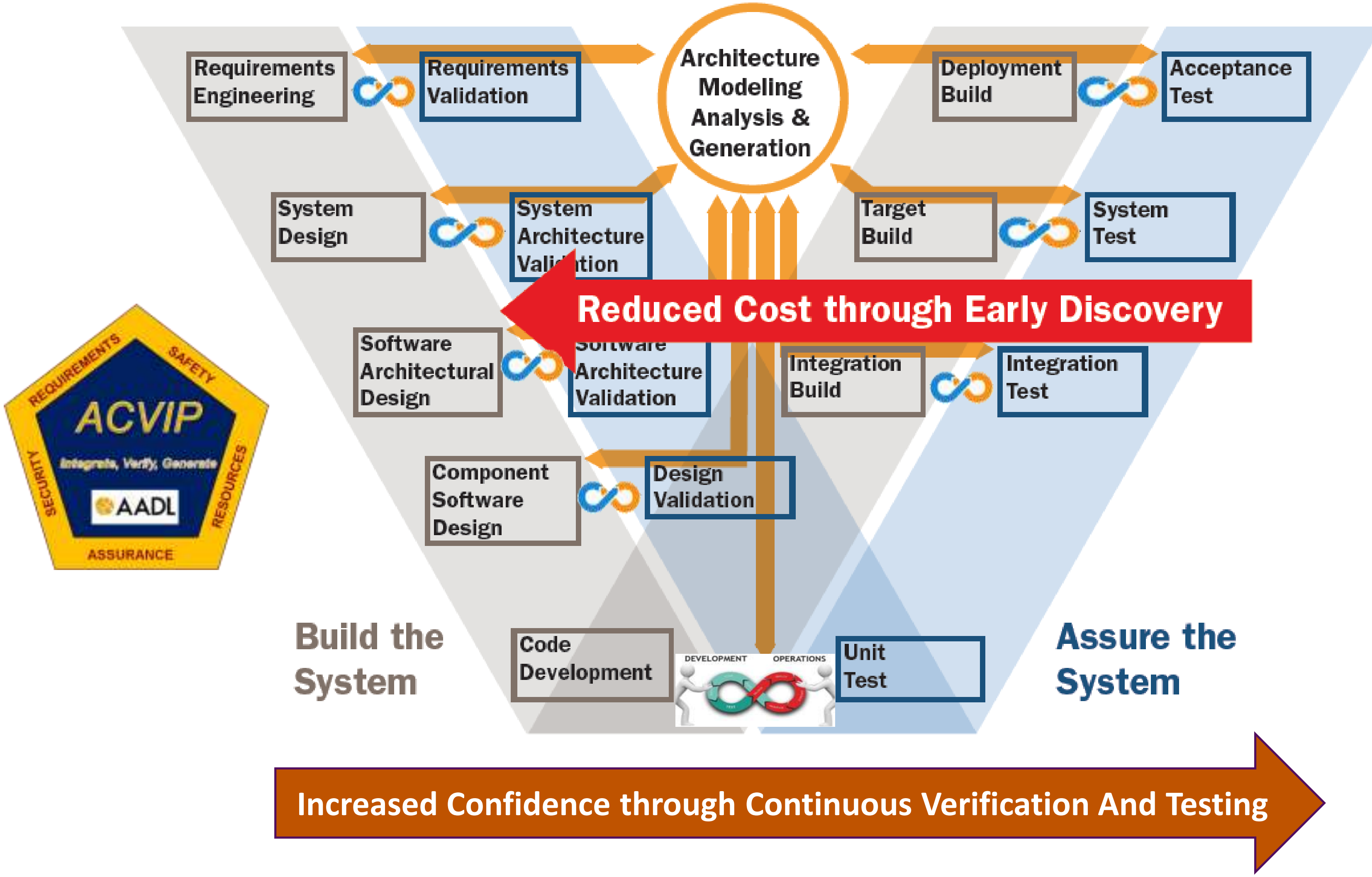
Needed Improvements:
- Analytics and models for estimating software costs
- Verification methods for requirements (including software assurance)
- Autonomy for exploring design options

Needed Improvements:
- Verification methods for system of system architectures → virtual integration
- "Deep" analysis and control of unverified code components
- Models and evidence for algorithmic correctness

Needed Improvements:
- Automated code analysis and repair (including malware)
- Human-machine safety, performance optimization and trust
- Mission assurance – quantified by analytics and architecture

Requirements   Development and Integration   Deployment and Operations   Sustainment

Benefits of Virtual System Integration & Continuous Lifecycle Assurance



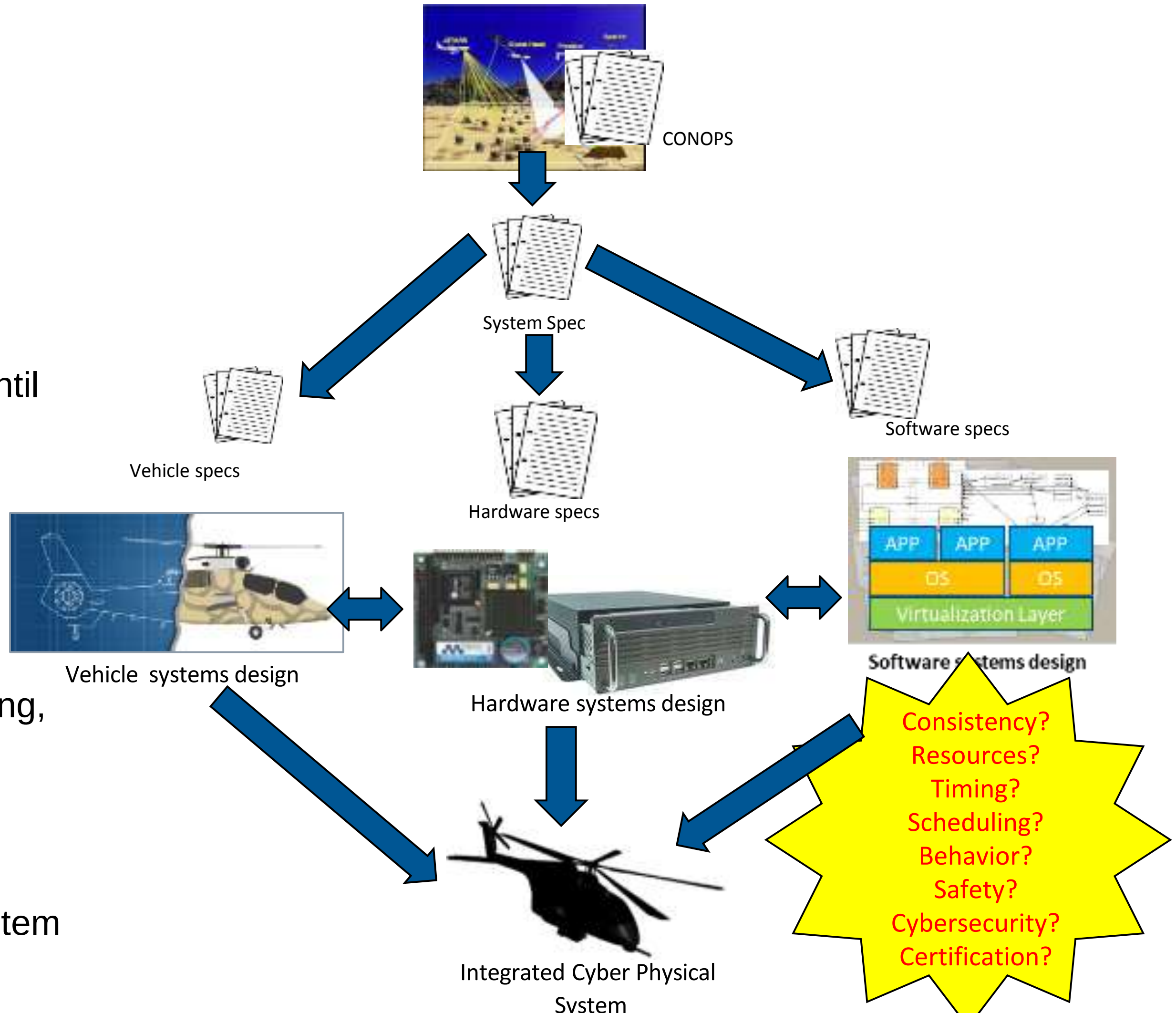
Systems Design & Challenges with Computing Systems Integration

Systems Design

- Involves multiple engineering disciplines
- Each requires different languages/tools/methods
- Most functionality deployed in software
- Software V&V does not begin until integration

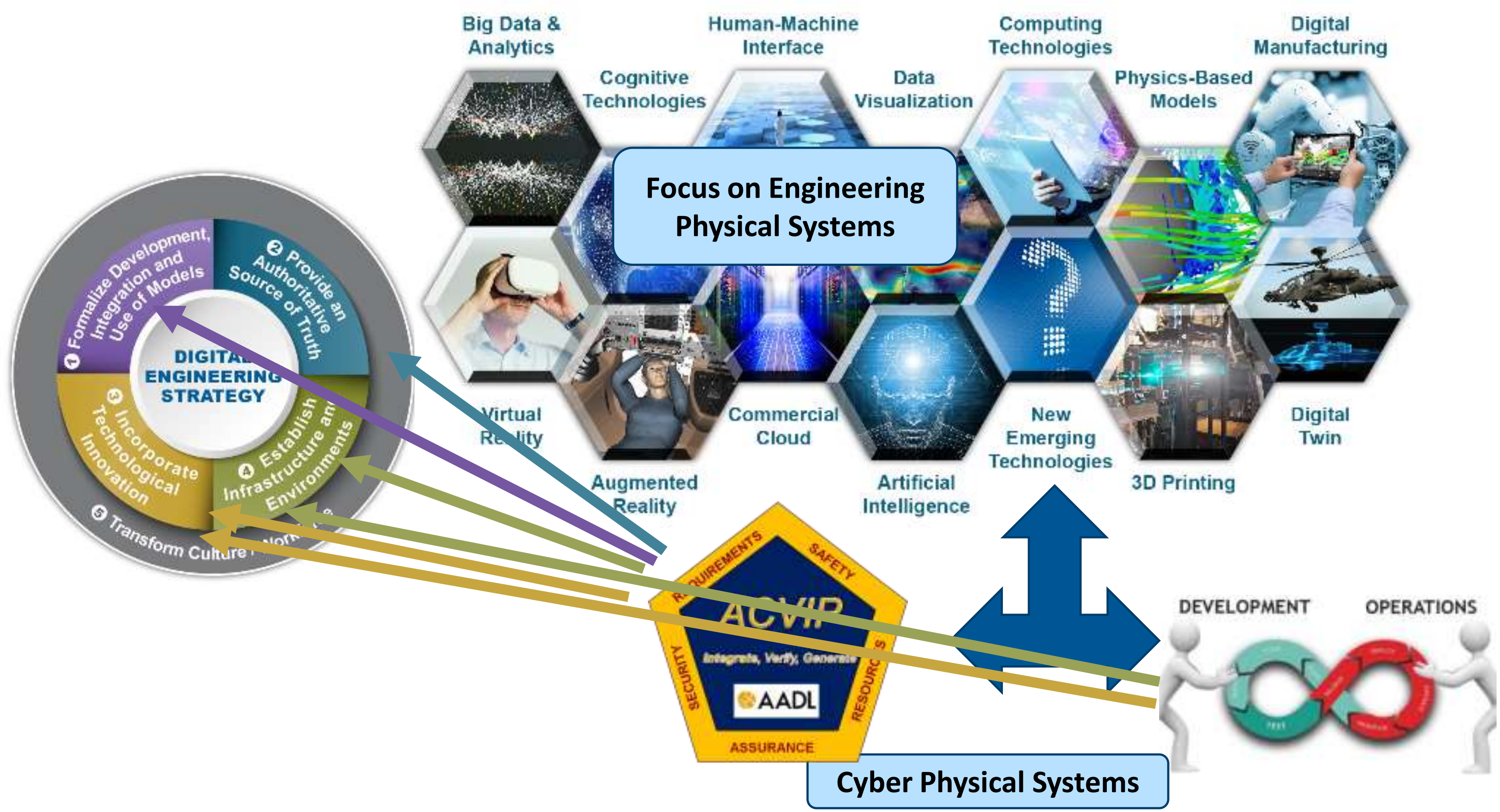
Design Challenges

- Maintaining consistency across design elements (units, data, messages, etc.)
- Inability to detect emergent side affects of limited resources, timing, scheduling
- Predicting interaction of requirements change during development & sustainment
- Qualifying and certifying the system



The growth in system complexity is being partially addressed by various MBSE tools and methods, but there remains challenges in integration and qualification of integration with software

DoD Digital Engineering Strategy: ACVIP as Key for Cyber Physical Systems



Summary

Safety Critical Embedded Software Systems are facing exponential growth in software development cost exceeding 70% of total system development cost

ACVIP is a set of technologies and practices that specifically have been designed to provide early detection and continuous verification throughout the life cycle

ACVIP is a key contributor to the DoD Digital Engineering Strategy