From Virtual System Integration to Incremental Lifecycle Assurance

Peter H. Feiler

Software Solutions Conference 2015

November 16–18, 2015

Software Engineering Institute | Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0003034

Agenda

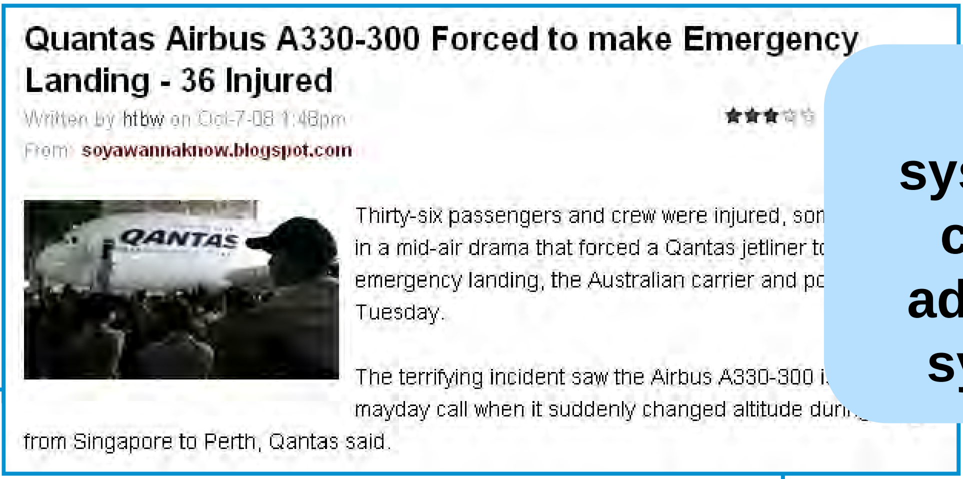
Challenges in Software Reliant Systems

Four Pillar Improvement Strategy

Virtual System Integration

Incremental Lifecycle Assurance

We Rely on Software for Safe Aircraft Operation



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

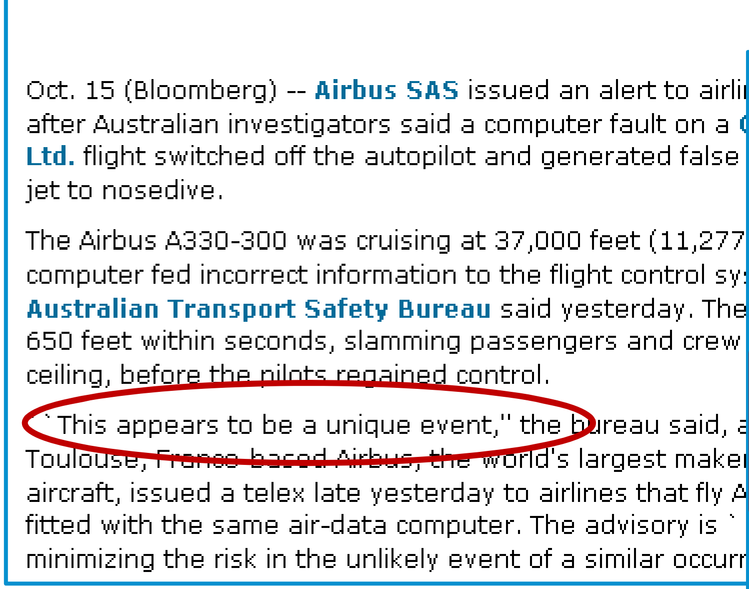
Written by htbw on Oct-7-08 1:48pm

From: soyawannaknow.blogspot.com

Thirty-six passengers and crew were injured, son in a mid-air drama that forced a Qantas jetliner to emergency landing, the Australian carrier and po Tuesday.

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

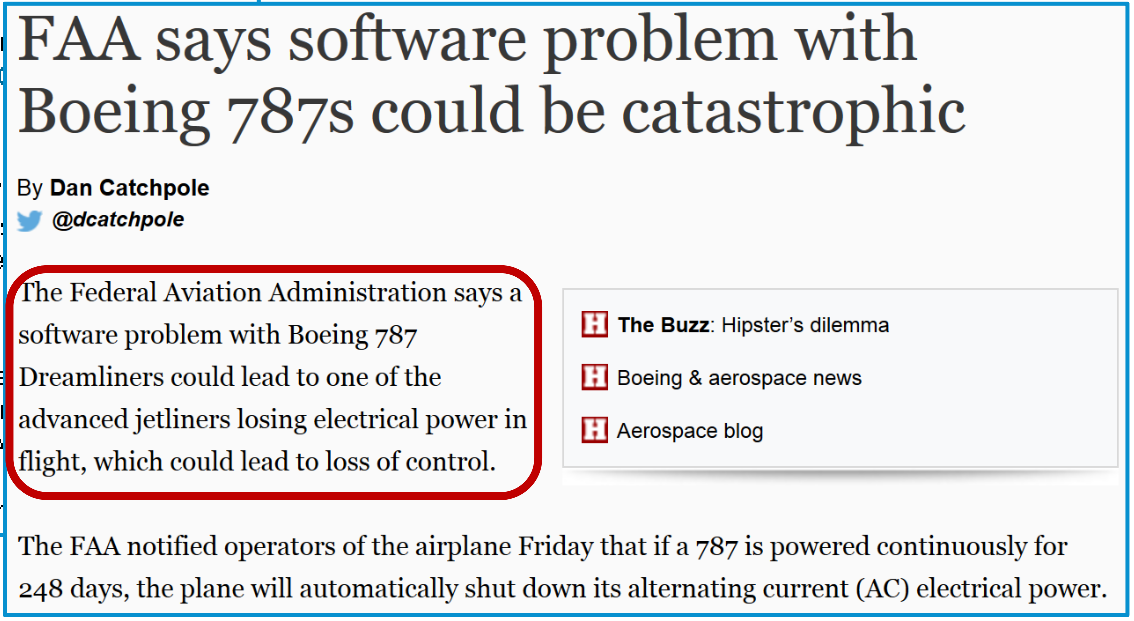
Embedded software systems introduce a new class of problems not addressed by traditional system safety analysis



Oct. 15 (Bloomberg) -- **Airbus SAS** issued an alert to airli after Australian investigators said a computer fault on a Q **Ltd.** flight switched off the autopilot and generated false jet to nosedive.

The Airbus A330-300 was cruising at 37,000 feet (11,277 computer fed incorrect information to the flight control sys **Australian Transport Safety Bureau** said yesterday. The 650 feet within seconds, slamming passengers and crew ceiling, before the pilots regained control.

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



FAA says software problem with Boeing 787s could be catastrophic

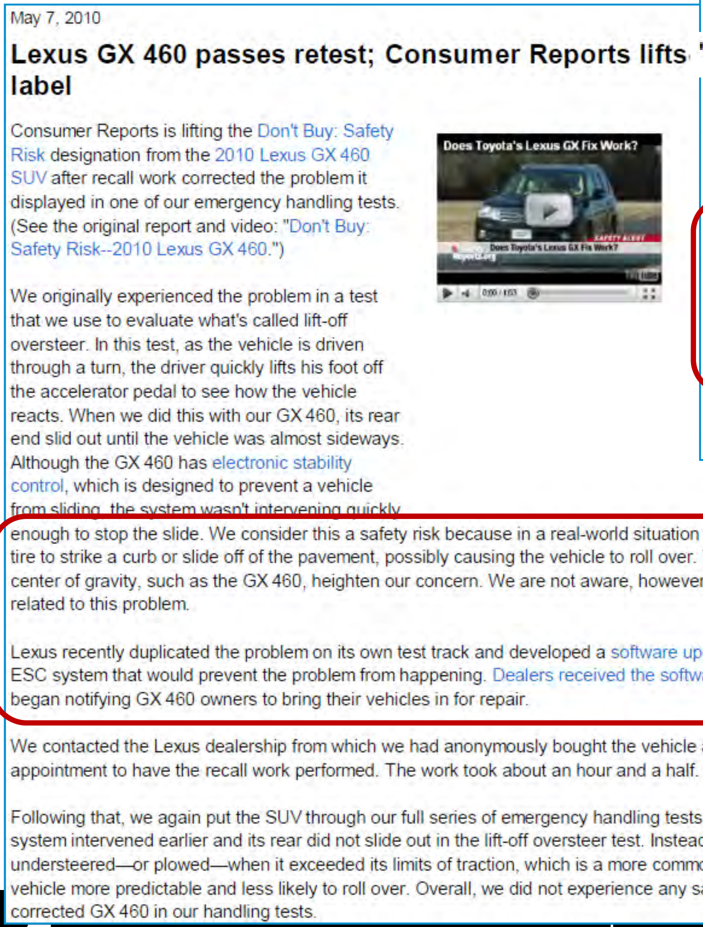
By **Dan Catchpole**

@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 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.

Software Problems not just in Aircraft



May 7, 2010

Lexus GX 460 passes retest; Consumer Reports lifts "Don't Buy" label

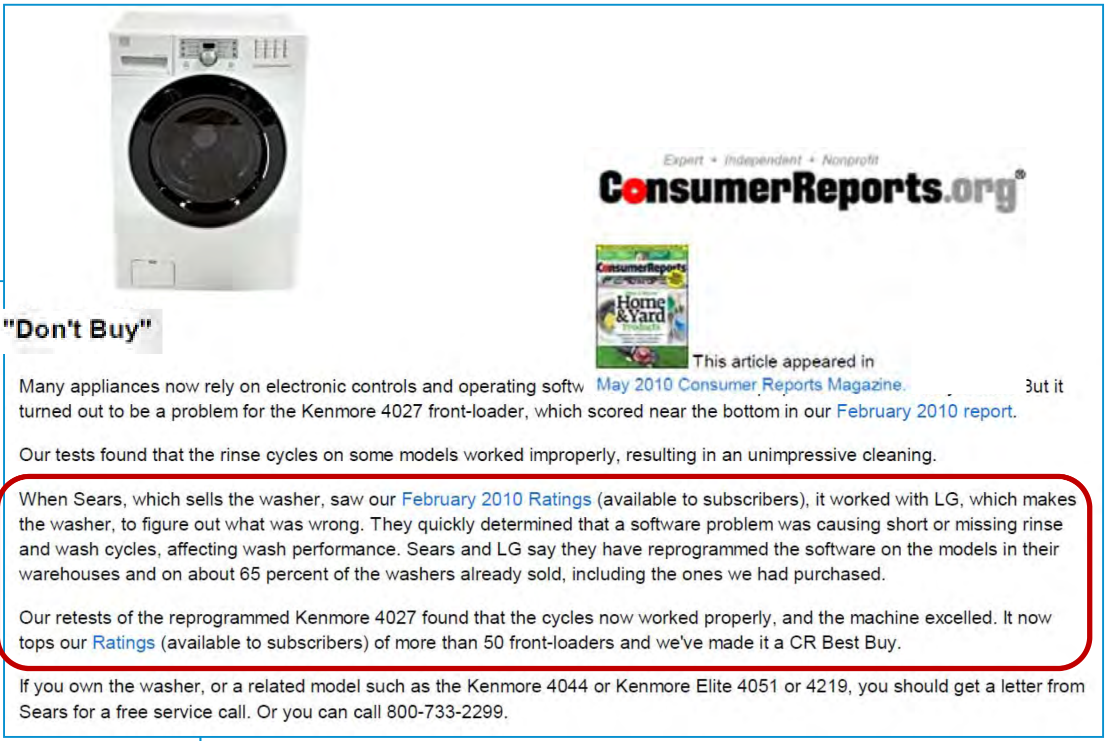
Consumer Reports is lifting the Don't Buy: Safety Risk designation from the 2010 Lexus GX 460 SUV after recall work corrected the problem it displayed in one of our emergency handling tests. (See the original report and video: "Don't Buy: Safety Risk--2010 Lexus GX 460.")

We originally experienced the problem in a test that we use to evaluate what's called lift-off oversteer. In this test, as the vehicle is driven through a turn, the driver quickly lifts his foot off the accelerator pedal to see how the vehicle reacts. When we did this with our GX 460, its rear end slid out until the vehicle was almost sideways. Although the GX 460 has electronic stability control, which is designed to prevent a vehicle from sliding, the system wasn't intervening quickly enough to stop the slide. We consider this a safety risk because in a real-world situation this could cause a rear tire to strike a curb or slide off of the pavement, possibly causing the vehicle to roll over. Tall vehicles with a high center of gravity, such as the GX 460, heighten our concern. We are not aware, however, of any reports of injury related to this problem.

Lexus recently duplicated the problem on its own test track and developed a software upgrade for the vehicle's ESC system that would prevent the problem from happening. Dealers received the software fix last week and began notifying GX 460 owners to bring their vehicles in for repair.

We contacted the Lexus dealership from which we had anonymously bought the vehicle and made an appointment to have the recall work performed. The work took about an hour and a half.

Following that, we again put the SUV through our full series of emergency handling tests. This time, the ESC system intervened earlier and its rear did not slide out in the lift-off oversteer test. Instead, the vehicle understeered—or plowed—when it exceeded its limits of traction, which is a more common result and makes the vehicle more predictable and less likely to roll over. Overall, we did not experience any safety concerns with the corrected GX 460 in our handling tests.



This article appeared in May 2010 Consumer Reports Magazine.

Many appliances now rely on electronic controls and operating softw 3ut it turned out to be a problem for the Kenmore 4027 front-loader, which scored near the bottom in our February 2010 report.

Our tests found that the rinse cycles on some models worked improperly, resulting in an unimpressive cleaning.

When Sears, which sells the washer, saw our February 2010 Ratings (available to subscribers), it worked with LG, which makes the washer, to figure out what was wrong. They quickly determined that a software problem was causing short or missing rinse and wash cycles, affecting wash performance. Sears and LG say they have reprogrammed the software on the models in their warehouses and on about 65 percent of the washers already sold, including the ones we had purchased.

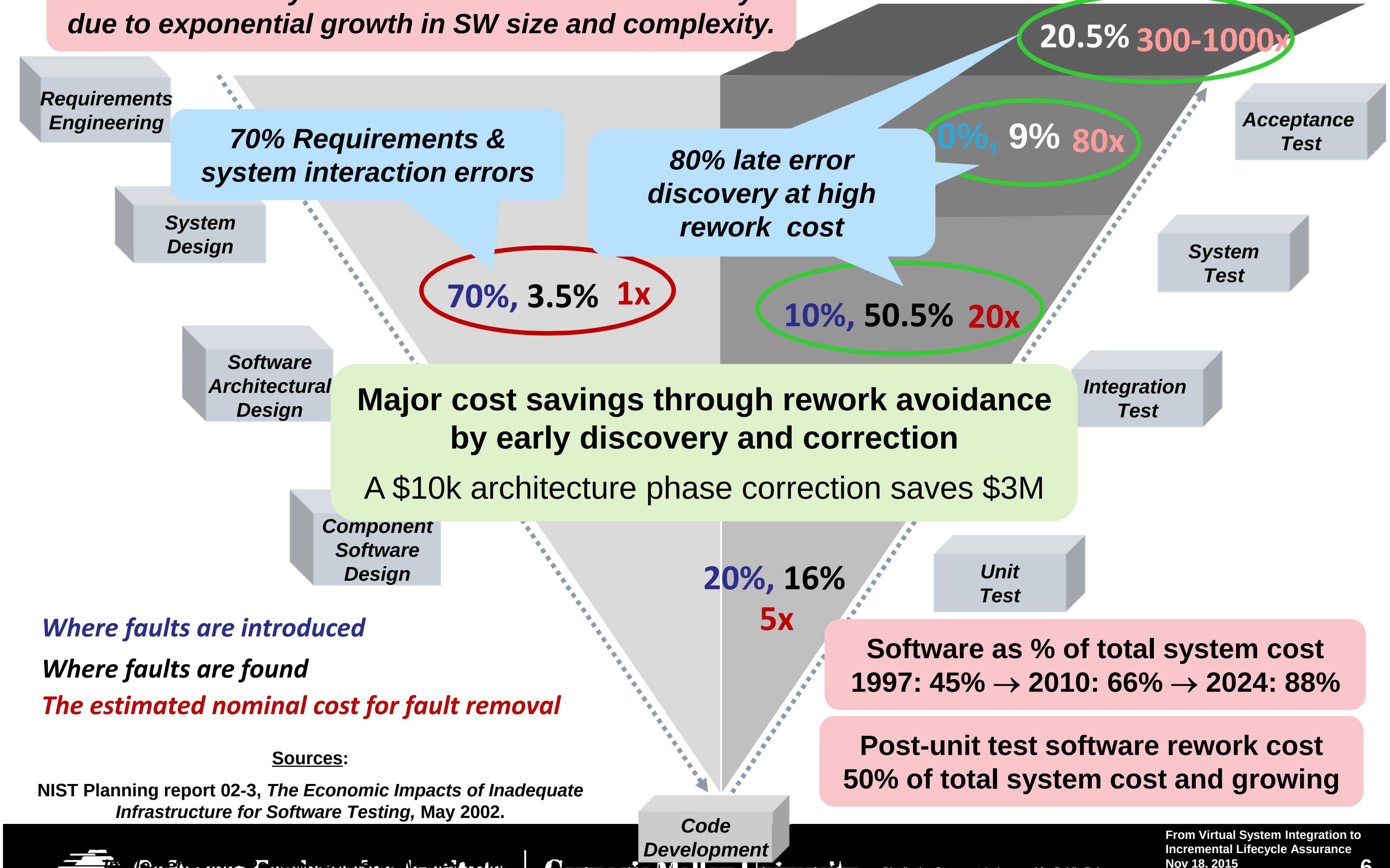
Our retests of the reprogrammed Kenmore 4027 found that the cycles now worked properly, and the machine excelled. It now tops our Ratings (available to subscribers) of more than 50 front-loaders and we've made it a CR Best Buy.

If you own the washer, or a related model such as the Kenmore 4044 or Kenmore Elite 4051 or 4219, you should get a letter from Sears for a free service call. Or you can call 800-733-2299.

How do you upgrade washing machine software?

High Fault Leakage Drives Major Increase in System Cost

Aircraft industry has reached limits of affordability due to exponential growth in SW size and complexity.



70% Requirements & system interaction errors

80% late error discovery at high rework cost

Major cost savings through rework avoidance by early discovery and correction

A $10k architecture phase correction saves $3M

Where faults are introduced

Where faults are found

The estimated nominal cost for fault removal

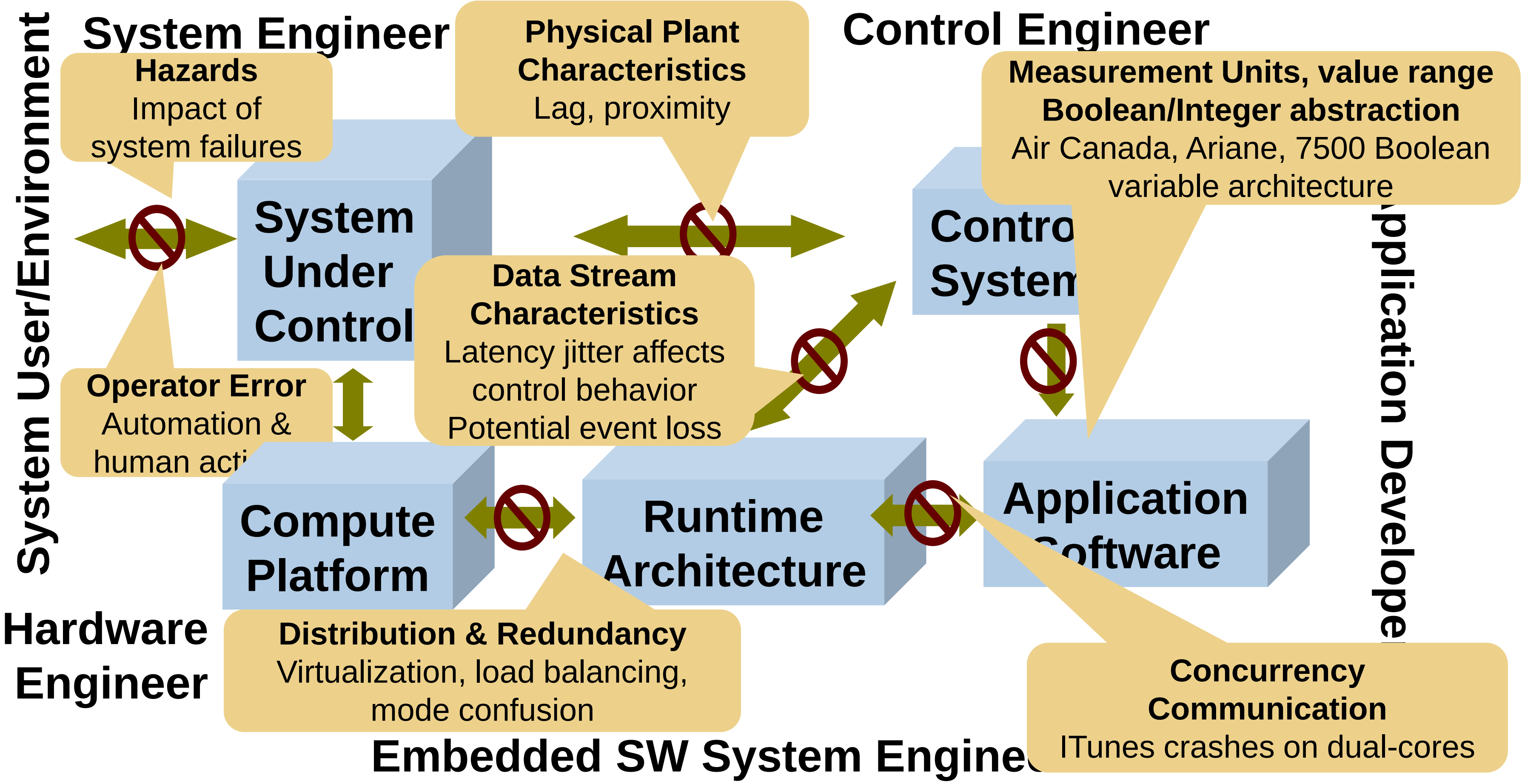
Software as % of total system cost
1997: 45% → 2010: 66% → 2024: 88%

Post-unit test software rework cost 50% of total system cost and growing

Sources:

NIST Planning report 02-3, *The Economic Impacts of Inadequate Infrastructure for Software Testing*, May 2002.

Mismatched Assumptions in System Interactions



System User/Environment

System Engineer

Hazards
Impact of
system failures

Physical Plant Characteristics
Lag, proximity

Control Engineer

Measurement Units, value range
Boolean/Integer abstraction
Air Canada, Ariane, 7500 Boolean
variable architecture

System Under Control

Contro System

pplication Develope

Data Stream Characteristics
Latency jitter affects
control behavior
Potential event loss

Operator Error
Automation &
human acti

Compute Platform

Runtime Architecture

Application Software

Hardware Engineer

Distribution & Redundancy
Virtualization, load balancing,
mode confusion

Concurrency
Communication
ITunes crashes on dual-cores

Embedded SW System Enginee

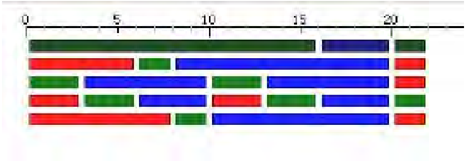
Embedded software system as major source of hazards

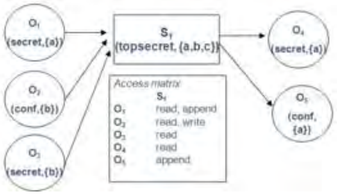
Why do system level failures still occur despite fault tolerance techniques being deployed in systems?

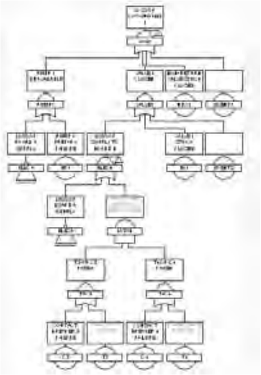
Model-based Engineering Pitfalls

The system

Inconsistency between independently developed analytical models

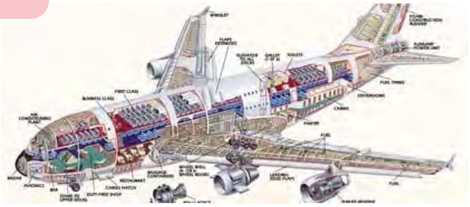






System models

Confidence that model reflects implementation



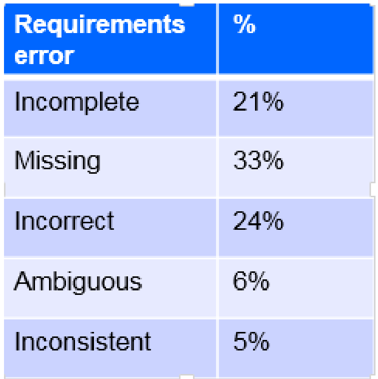
System implementation

This aircraft industry experience has led to the System Architecture Virtual Integration (SAVI) initiative

Awareness of Requirement Quality

Textual requirement quality statistics

- Current requirement engineering practice relies on stakeholders traceability and document reviews resulting in high rate of requirement change

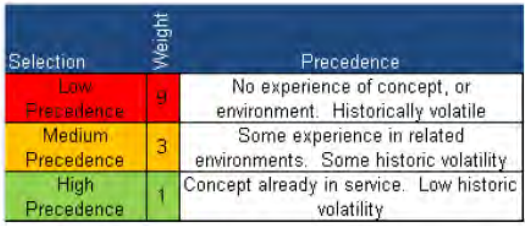


Requirements error	%
Incomplete	21%
Missing	33%
Incorrect	24%
Ambiguous	6%
Inconsistent	5%

NIST Study

Managed awareness of requirement uncertainty reduces requirement changes by 50%

- 80% of requirement changes from development team
- Expert assessment of change uncertainty
- Focus on high uncertainty and high importance areas
- Engineer for inherent variability



Selection	Weight	Precedence
Low Precedence	9	No experience of concept, or environment. Historically volatile
Medium Precedence	3	Some experience in related environments. Some historic volatility
High Precedence	1	Concept already in service. Low historic volatility

Figure 8. Precedence measurement scale

Rolls Royce Study

Agenda

Challenges in Software Reliant Systems

Four Pillar Improvement Strategy

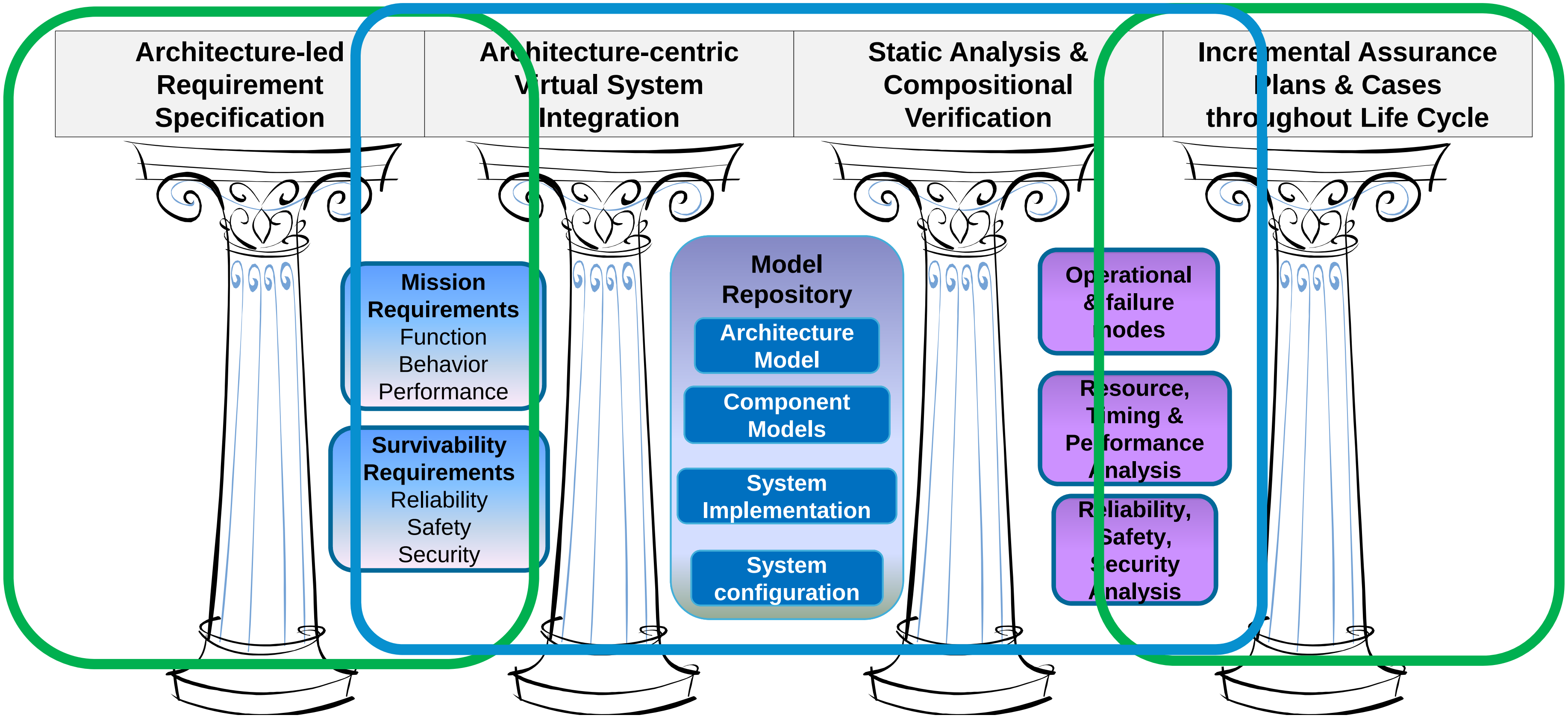
Virtual System Integration

Incremental Lifecycle Assurance

Assurance & Qualification Improvement Strategy

Assurance: <u>Sufficient</u> <u>evidence</u> that a <u>system implementation</u> meets <u>system requirements</u>

2010 SEI Study for AMRDEC
Aviation Engineering Directorate



Architecture-led Requirement Specification

Architecture-centric Virtual System Integration

Static Analysis & Compositional Verification

Incremental Assurance Plans & Cases throughout Life Cycle

Mission Requirements
Function
Behavior
Performance

Survivability Requirements
Reliability
Safety
Security

Model Repository

Architecture Model

Component Models

System Implementation

System configuration

Operational & failure modes

Resource, Timing & Performance Analysis

Reliability, Safety, Security Analysis

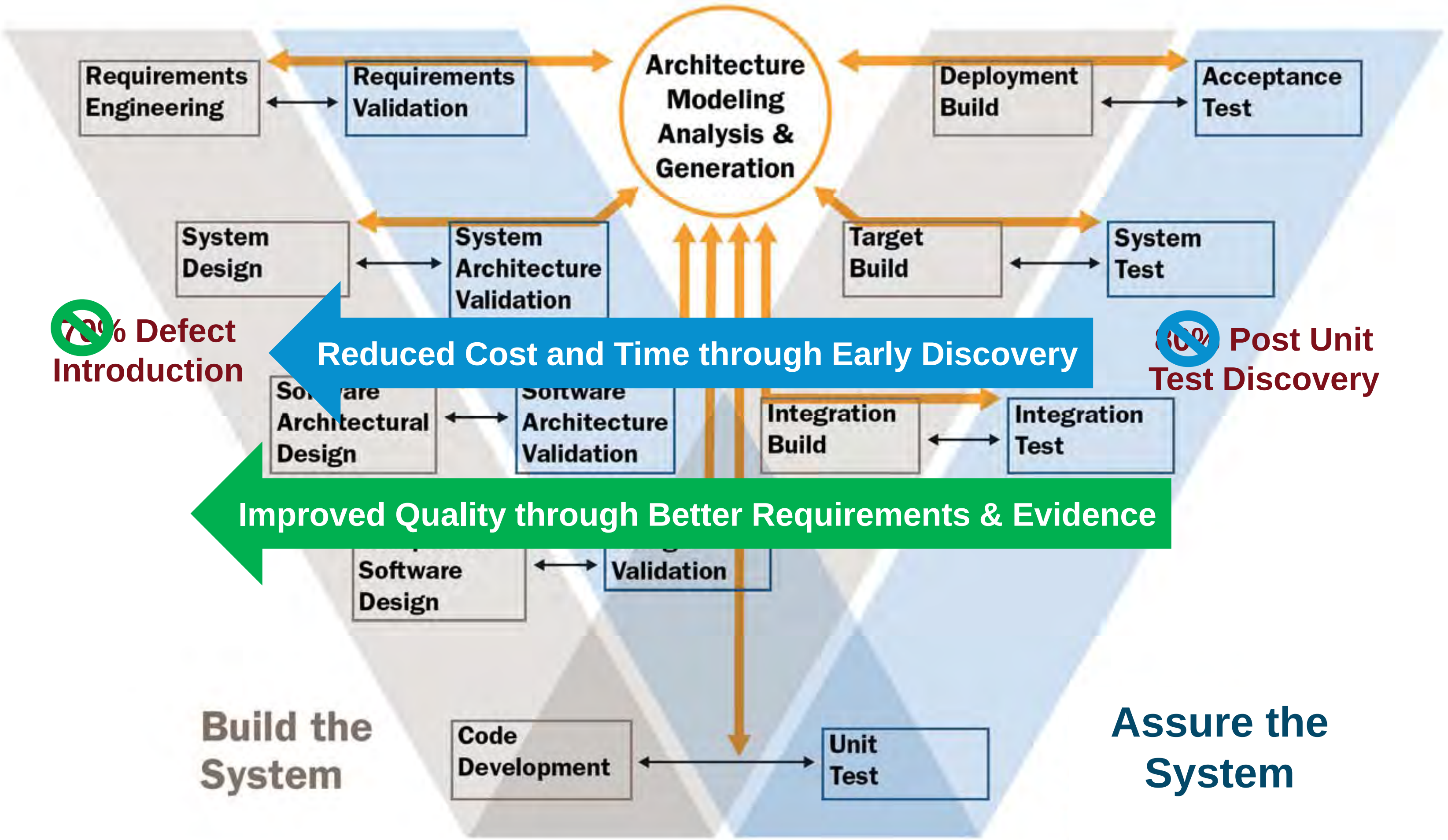
Early Problem Discovery through Virtual System Integration & Analysis

Improved Assurance through Better Requirements & Automated Verification

Software Engineering Institute | Carnegie Mellon University

Distribution Statement A: Approved for Public Release;
Distribution is Unlimited

Improved Cost, Time and Quality



70% Defect Introduction

Reduced Cost and Time through Early Discovery

80% Post Unit Test Discovery

Improved Quality through Better Requirements & Evidence

Agenda

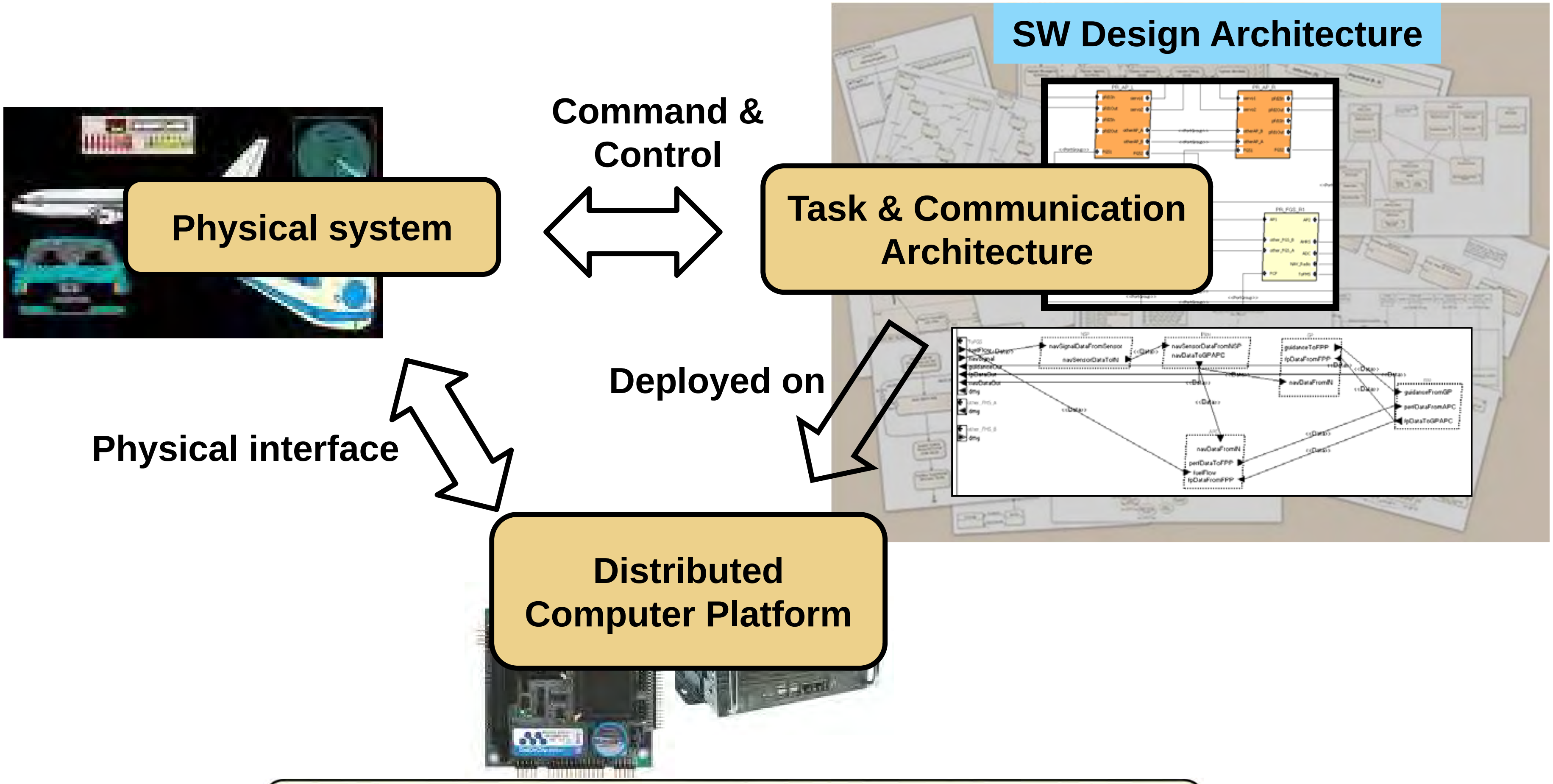
Challenges in Software Reliant Systems

Four Pillar Improvement Strategy

Virtual System Integration

Incremental Lifecycle Assurance

SAE Architecture Analysis & Design Language (AADL) to the Rescue



SW Design Architecture

Command & Control

Physical system

Task & Communication Architecture

Deployed on

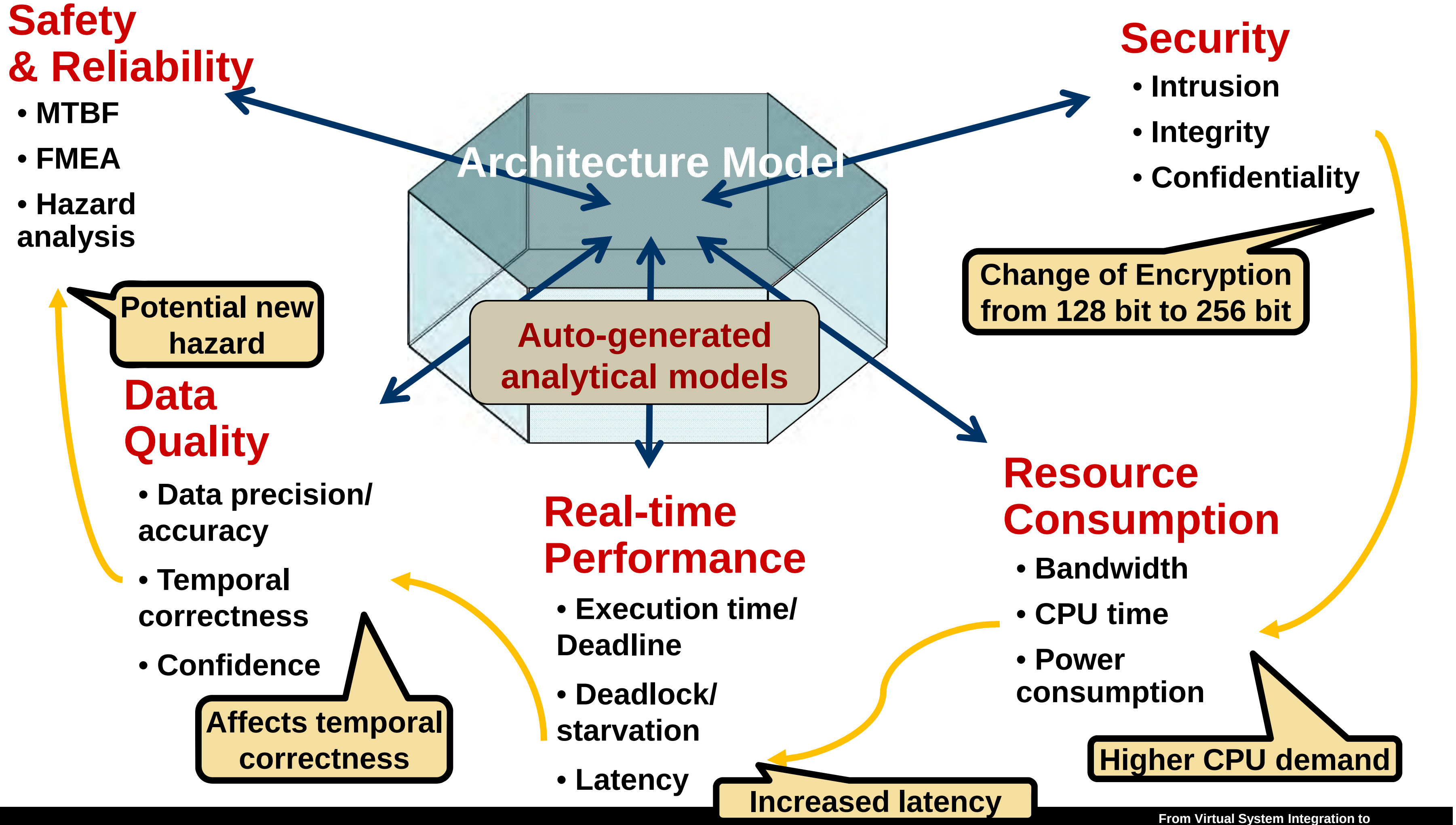
Physical interface

Distributed Computer Platform

AADL Addresses Increasing Interaction Complexity and Mismatched Assumptions

Analysis of Virtually Integrated Software Systems

Single Annotated Architecture Model Addresses Impact Across Operational Quality Attributes



Safety & Reliability

- MTBF
- FMEA
- Hazard analysis

Security

- Intrusion
- Integrity
- Confidentiality

Architecture Model

Auto-generated analytical models

Potential new hazard

Change of Encryption from 128 bit to 256 bit

Data Quality

- Data precision/ accuracy
- Temporal correctness
- Confidence

Real-time Performance

- Execution time/ Deadline
- Deadlock/ starvation
- Latency

Resource Consumption

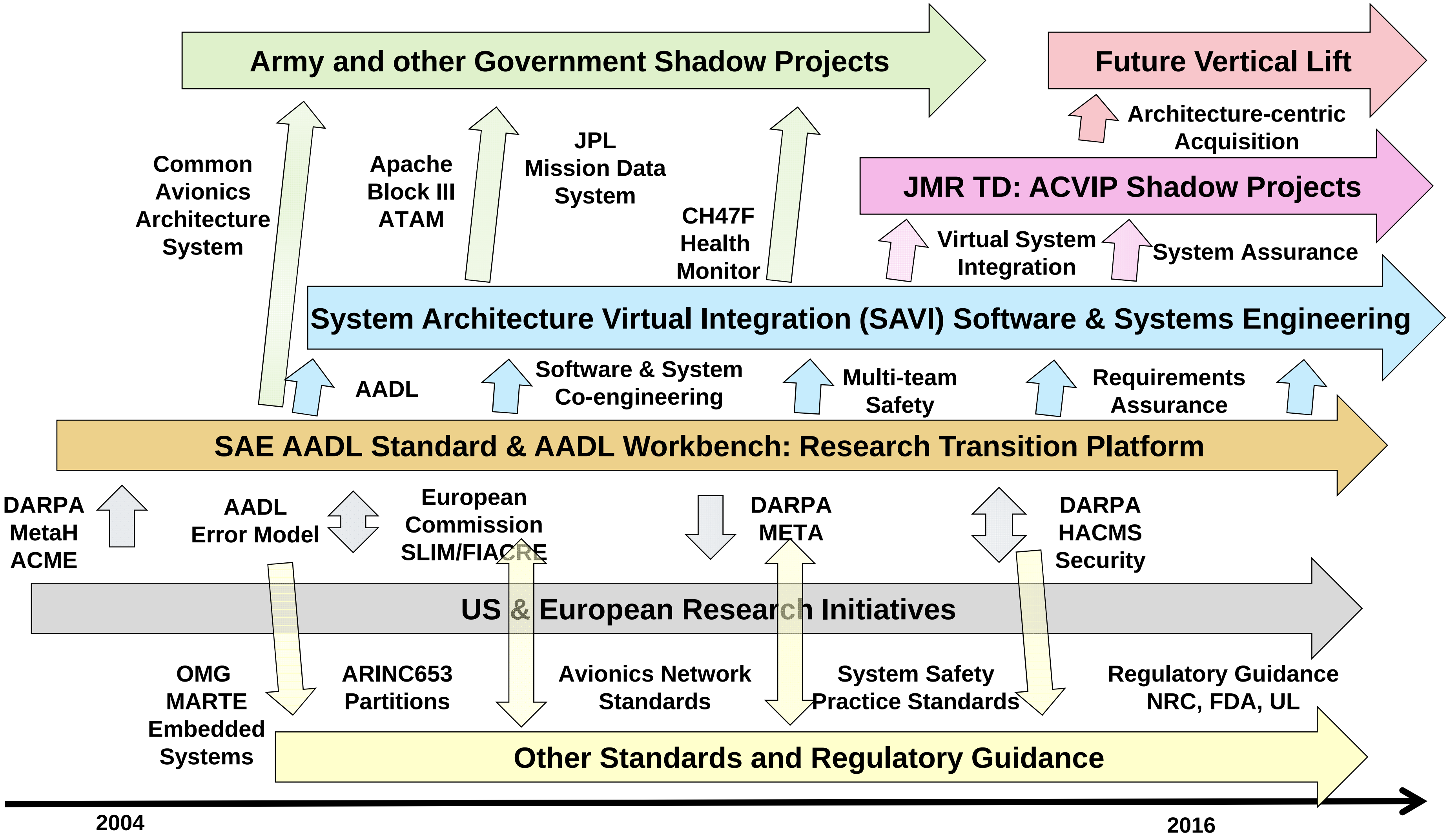
- Bandwidth
- CPU time
- Power consumption

Affects temporal correctness

Increased latency

Higher CPU demand

Towards an Architecture-Centric Virtual Integration Practice (ACVIP)



Army and other Government Shadow Projects

Future Vertical Lift

Architecture-centric Acquisition

JMR TD: ACVIP Shadow Projects

Common Avionics Architecture System

Apache Block III ATAM

JPL Mission Data System

CH47F Health Monitor

Virtual System Integration

System Assurance

System Architecture Virtual Integration (SAVI) Software & Systems Engineering

AADL

Software & System Co-engineering

Multi-team Safety

Requirements Assurance

SAE AADL Standard & AADL Workbench: Research Transition Platform

DARPA MetaH ACME

AADL Error Model

European Commission SLIM/FIACRE

DARPA META

DARPA HACMS Security

US & European Research Initiatives

OMG MARTE Embedded Systems

ARINC653 Partitions

Avionics Network Standards

System Safety Practice Standards

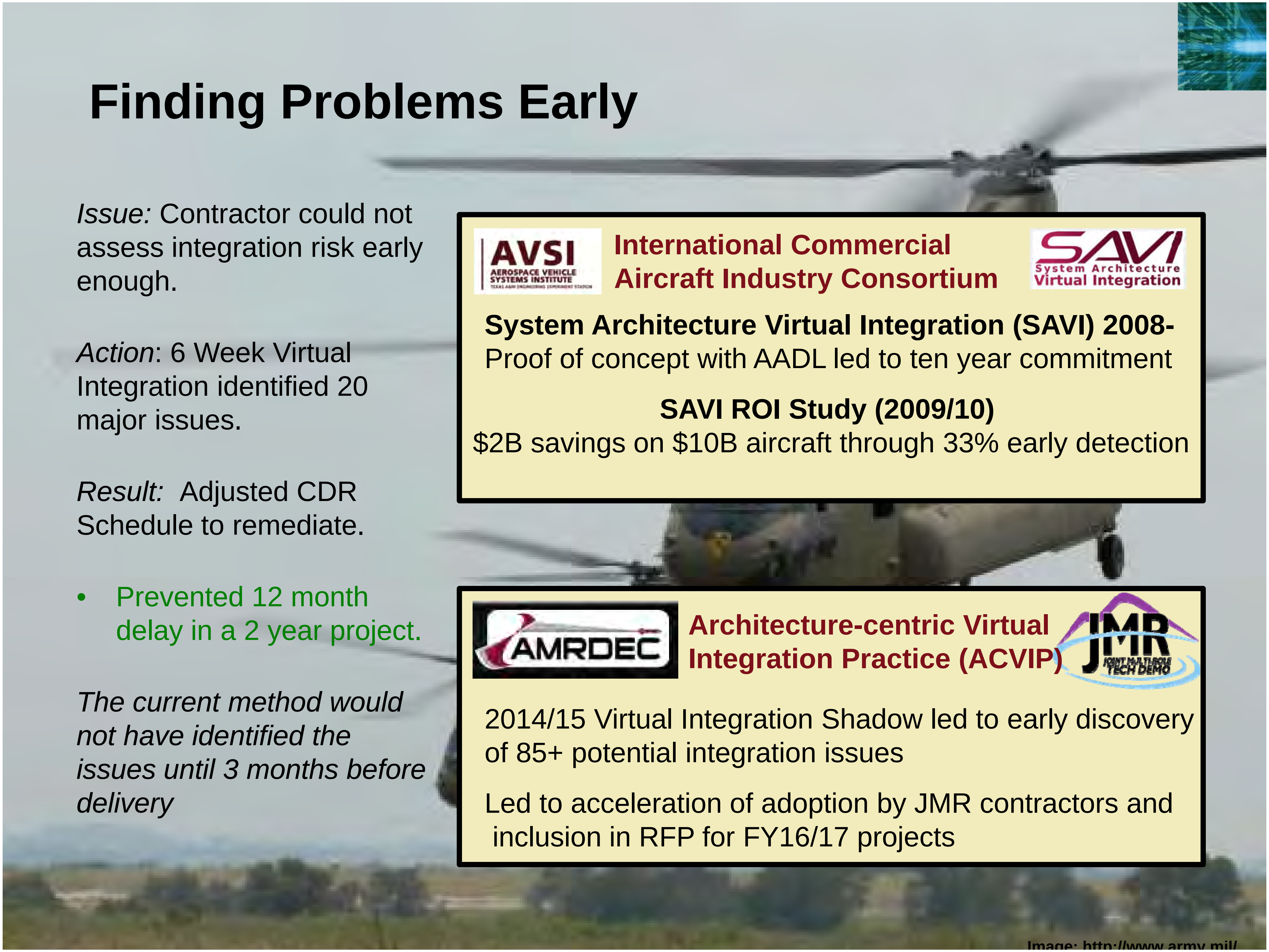
Regulatory Guidance NRC, FDA, UL

Other Standards and Regulatory Guidance

2004

2016

Finding Problems Early



Issue: Contractor could not assess integration risk early enough.

Action: 6 Week Virtual Integration identified 20 major issues.

Result: Adjusted CDR Schedule to remediate.

- Prevented 12 month delay in a 2 year project.

The current method would not have identified the issues until 3 months before delivery

International Commercial Aircraft Industry Consortium

System Architecture Virtual Integration (SAVI) 2008-
Proof of concept with AADL led to ten year commitment

SAVI ROI Study (2009/10)
$2B savings on $10B aircraft through 33% early detection

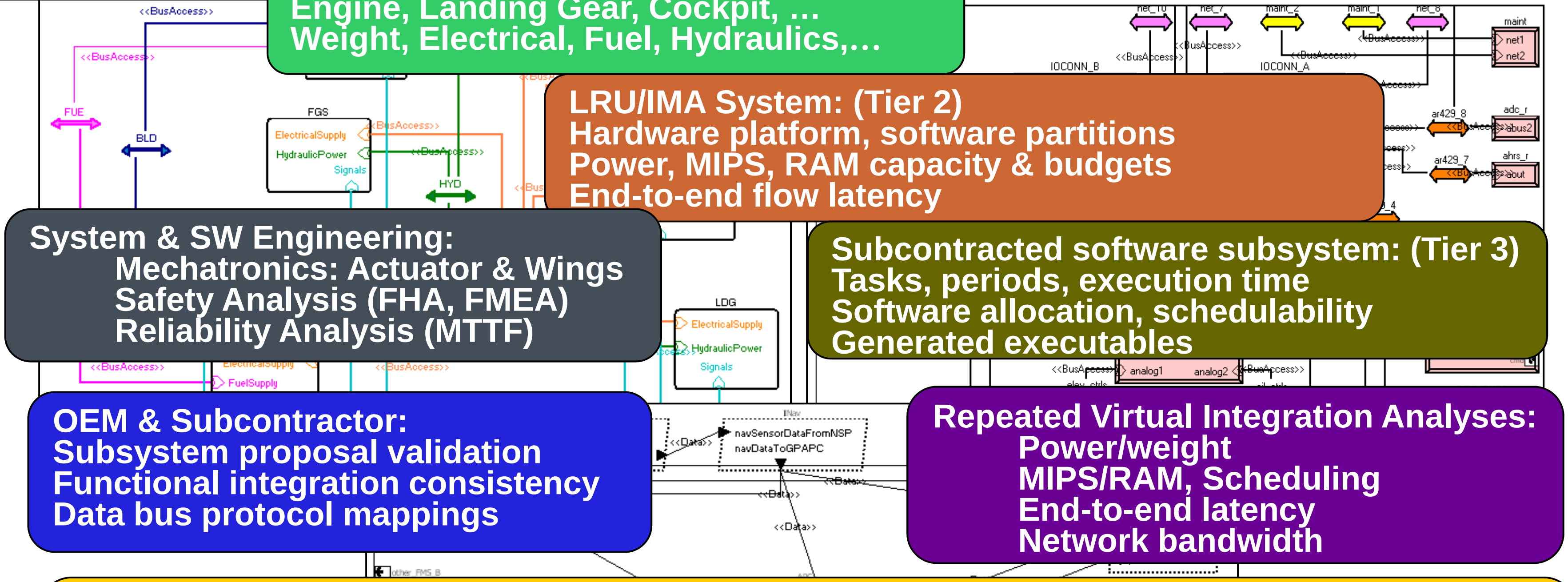
Architecture-centric Virtual Integration Practice (ACVIP)

2014/15 Virtual Integration Shadow led to early discovery of 85+ potential integration issues

Led to acceleration of adoption by JMR contractors and inclusion in RFP for FY16/17 projects

Image: http://www.army.mil/

Incremental Multi-Tier Assurance in SAVI



Aircraft: (Tier 0)

Aircraft system: (Tier 1)
Engine, Landing Gear, Cockpit, …
Weight, Electrical, Fuel, Hydraulics,…

LRU/IMA System: (Tier 2)
Hardware platform, software partitions
Power, MIPS, RAM capacity & budgets
End-to-end flow latency

System & SW Engineering:
Mechatronics: Actuator & Wings
Safety Analysis (FHA, FMEA)
Reliability Analysis (MTTF)

Subcontracted software subsystem: (Tier 3)
Tasks, periods, execution time
Software allocation, schedulability
Generated executables

OEM & Subcontractor:
Subsystem proposal validation
Functional integration consistency
Data bus protocol mappings

Repeated Virtual Integration Analyses:
Power/weight
MIPS/RAM, Scheduling
End-to-end latency
Network bandwidth

Proof of Concept Demonstration and Transition by Aerospace industry initiative

- **Architecture-centric model-based software and system engineering**
- **Architecture-centric model-based acquisition and development process**
- **Multi notation, multi team model repository & standardized model interchange**

- Multi-tier system & software architecture (in AADL)
- Incremental end-to-end verification of system properties

Automated FMEA Experience

Failure Modes and Effects Analyses are rigorous and comprehensive reliability and safety design evaluations

- Required by industry standards and Government policies
- When performed manually are usually done once due to cost and schedule
- If automated allows for
  - multiple iterations from conceptual to detailed design
  - Tradeoff studies and evaluation of alternatives
  - Early identification of potential problems

Largest analysis of satellite to date consists of 26,000 failure modes

- Includes detailed model of satellite bus
- 20 states perform failure mode
- Longest failure mode sequences have 25 transitions (i.e., 25 effects)

Myron Hecht, Aerospace Corp.
Safety Analysis for JPL, member of DO-178C committee

Agenda

Challenges in Software Reliant Systems

Four Pillar Improvement Strategy

Virtual System Integration

Incremental Lifecycle Assurance

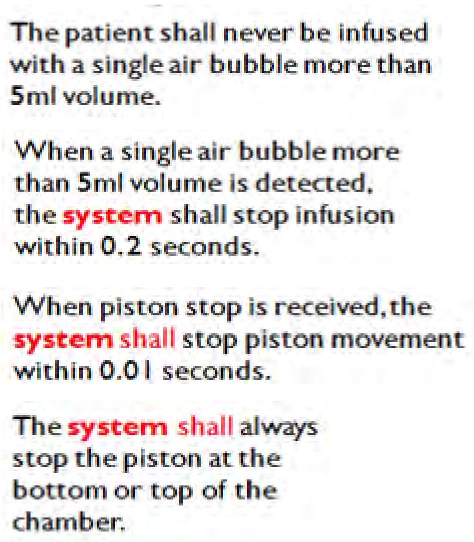
Incremental Lifecycle Assurance Objectives

Measurably improve critical system assurance through

- Better coverage and managed uncertainty
- Incremental analytical verification throughout lifecycle
- Focus on high payoff areas

Requirements & Architecture Design Constraints

Textual Requirements for a Patient Therapy System



The patient shall never be infused with a single air bubble more than 5ml volume.

When a single air bubble more than 5ml volume is detected, the system shall stop infusion within 0.2 seconds.

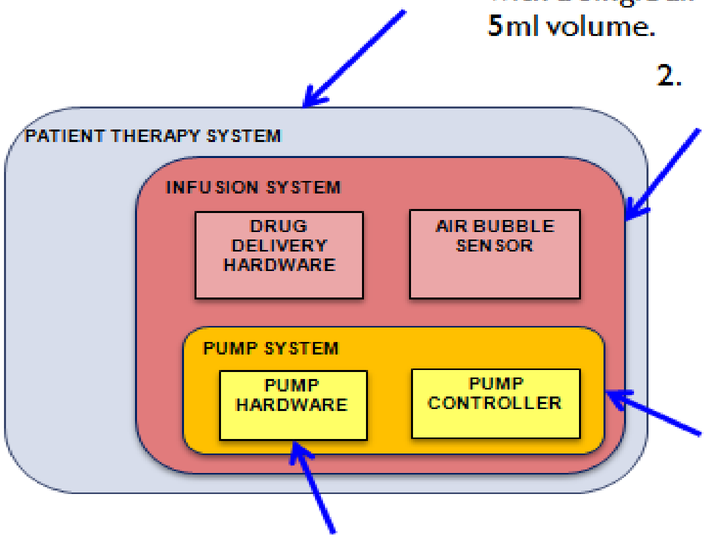
When piston stop is received, the system shall stop piston movement within 0.01 seconds.

The system shall always stop the piston at the bottom or top of the chamber.

Same Requirements Mapped to an Architecture Model

1. The patient shall never be infused with a single air bubble more than 5ml volume.

2. When a single air bubble more than 5ml volume is detected, the system shall stop infusion within 0.2 seconds.



4. The system shall always stop the piston at the bottom or top of the chamber.

3. When piston stop is received, the system shall stop piston movement within 0.01 seconds.

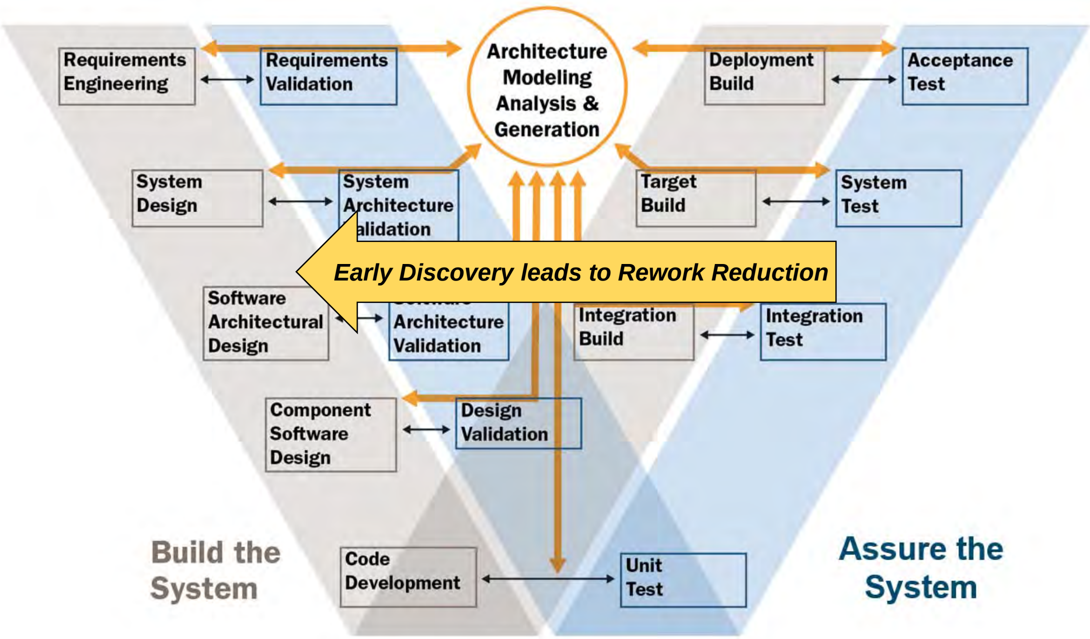
Importance of understanding system boundary

We have effectively specified a system partial architecture

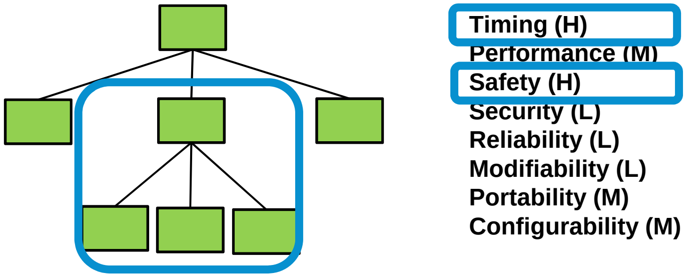
U Minnesota Study

Three Dimensions of Incremental Assurance

Incremental assurance through virtual system integration for early discovery

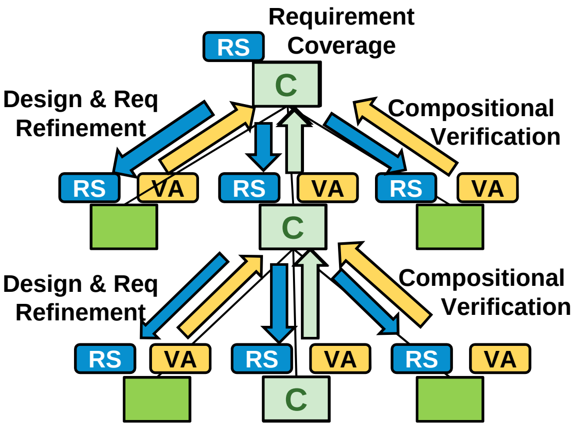


Priority focused architecture design exploration for high payoff



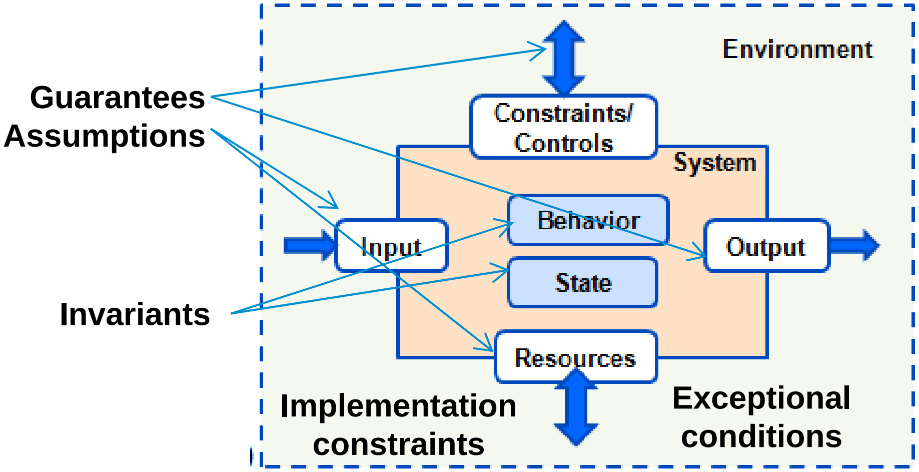
Timing (H)
Performance (M)
Safety (H)
Security (L)
Reliability (L)
Modifiability (L)
Portability (M)
Configurability (M)

Compositional verification and partitions to limit assurance impact

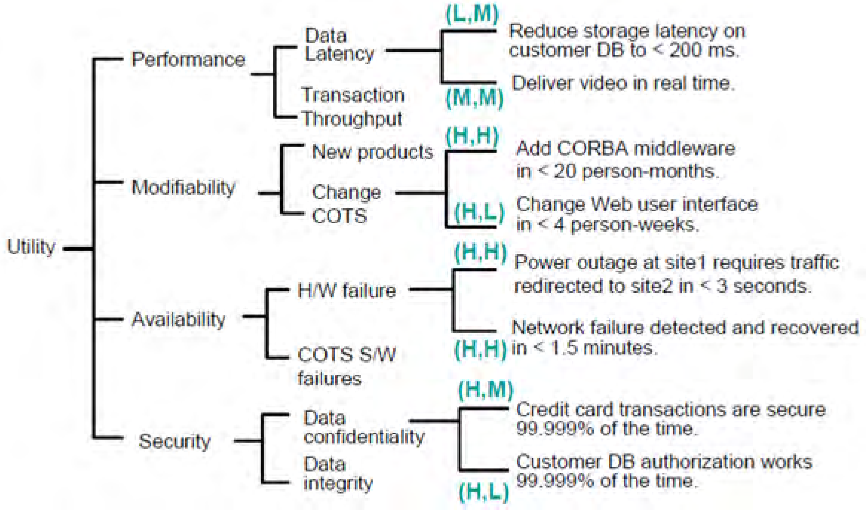


Three Dimensions of Requirement Coverage

System interactions, state, behavior


Guarantees
Assumptions
Invariants
Environment
Constraints/ Controls
System
Input
Behavior
State
Output
Resources
Implementation constraints
Exceptional conditions

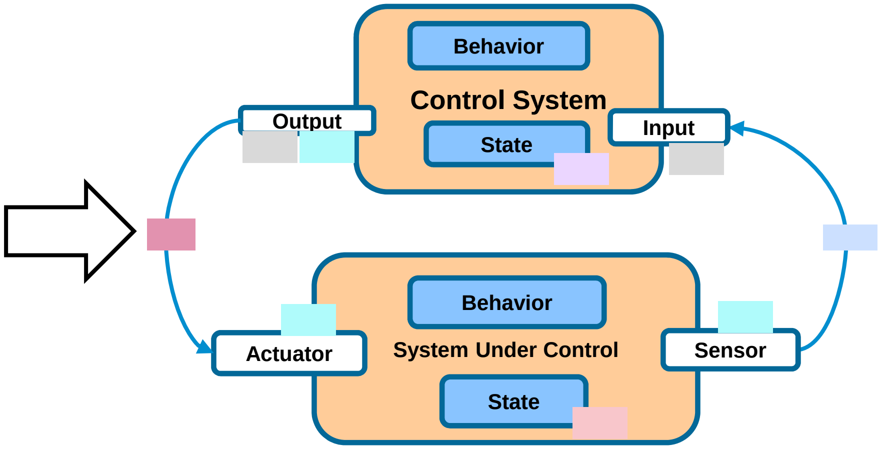
Design & operational quality attributes


Utility
Performance
Data Latency
(L,M) Reduce storage latency on customer DB to < 200 ms.
Transaction Throughput
(M,M) Deliver video in real time.
Modifiability
New products
(H,H) Add CORBA middleware in < 20 person-months.
Change COTS
(H,L) Change Web user interface in < 4 person-weeks.
Availability
H/W failure
(H,H) Power outage at site1 requires traffic redirected to site2 in < 3 seconds.
COTS S/W failures
(H,H) Network failure detected and recovered in < 1.5 minutes.
Security
Data confidentiality
(H,M) Credit card transactions are secure 99.999% of the time.
Data integrity
(H,L) Customer DB authorization works 99.999% of the time.

Fault impact & contributors

Omission errors	Commission errors
Value errors	Sequence errors
Timing errors	Replication errors
Rate errors	Concurrency errors
Authentication errors	Authorization errors

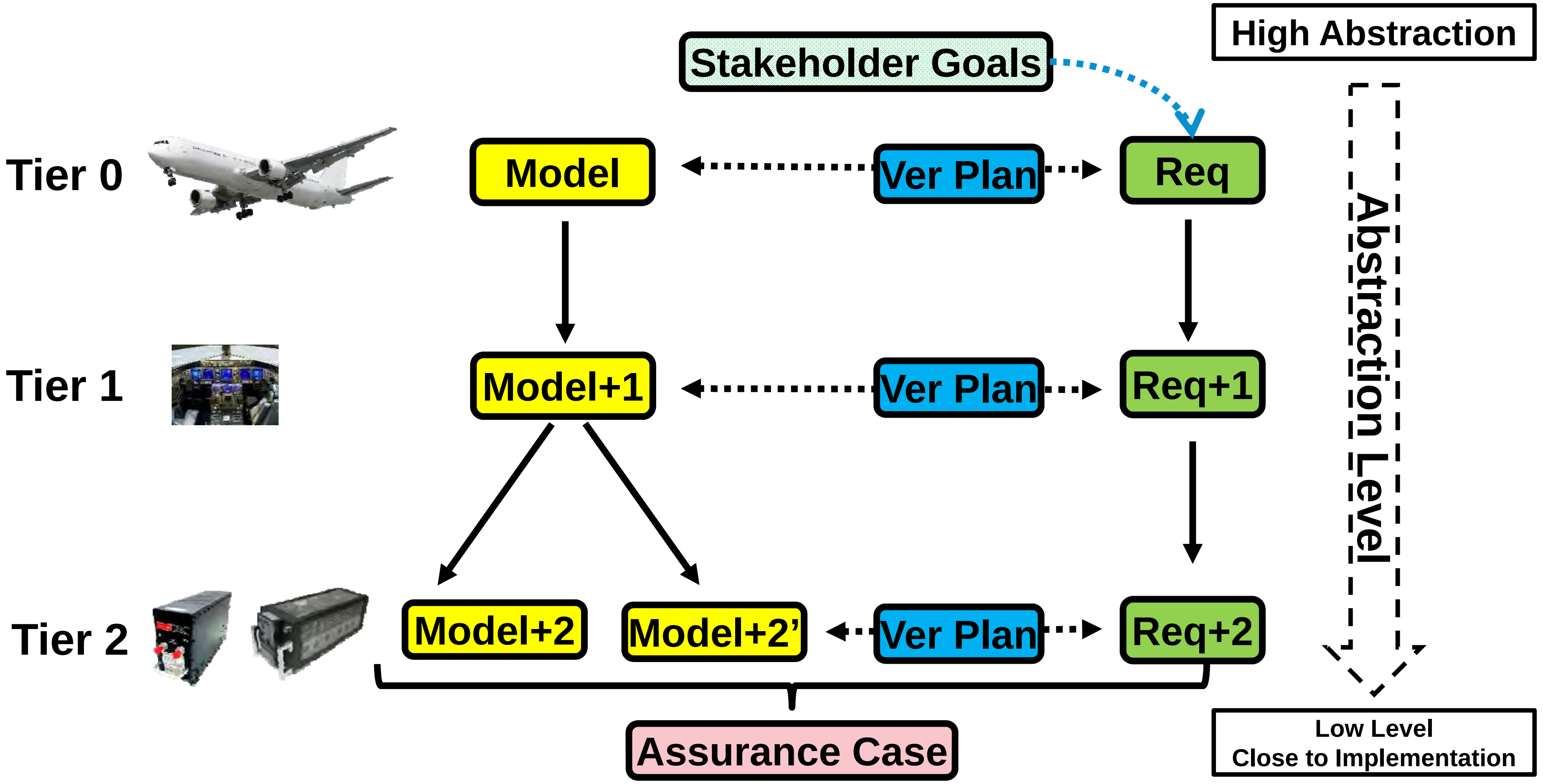
Fault Propagation Ontology


Behavior
Control System
Output
State
Input
Behavior
Actuator
System Under Control
Sensor
State

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

Automated Incremental Assurance Workbench

Identify Assurance Hotspots throughout Lifecycle



High Abstraction

Stakeholder Goals

Tier 0

Model

Ver Plan

Req

Tier 1

Model+1

Ver Plan

Req+1

Tier 2

Model+2

Model+2’

Ver Plan

Req+2

Abstraction Level

Assurance Case

Low Level
Close to Implementation

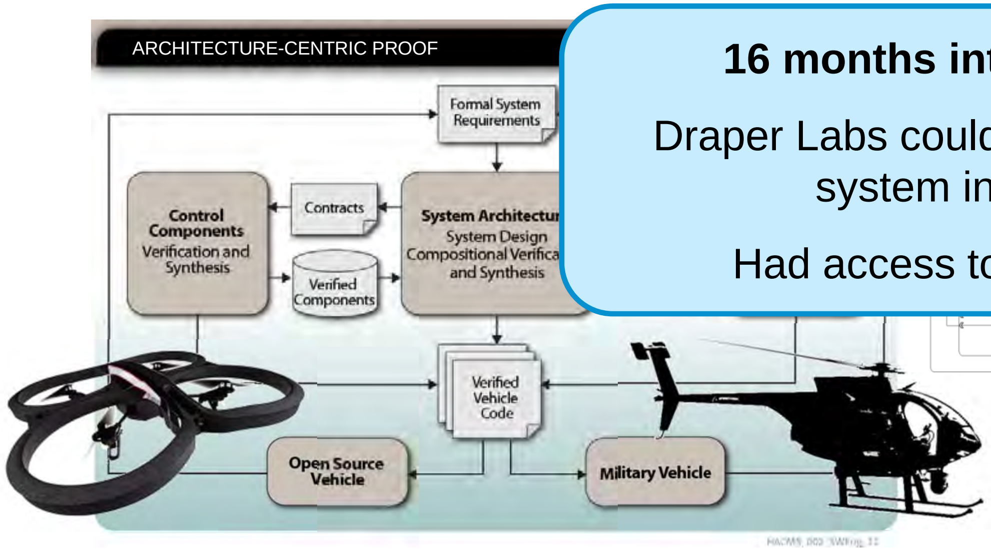
Contract-based Compositional Verification

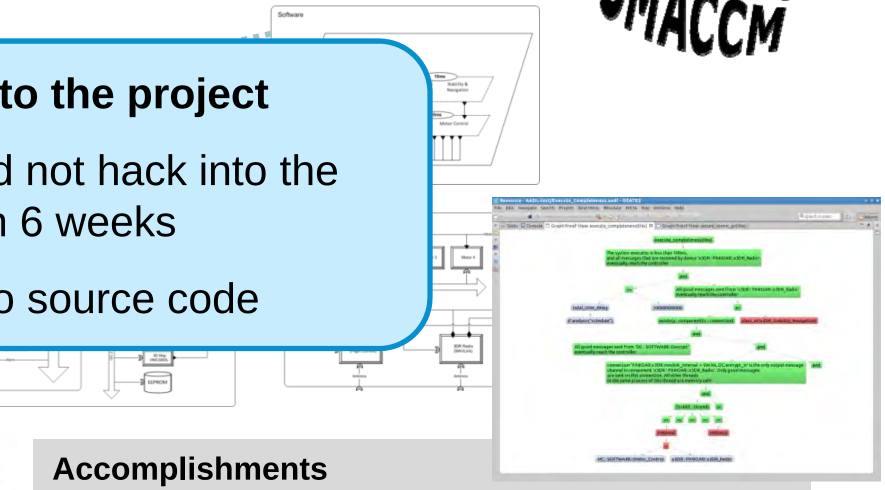
Secure Mathematically-Assured Composition of Control Models

Key Problem

Many vulnerabilities occur at component interfaces. How can we use formal methods to detect these vulnerabilities and build provably secure systems?

TA4 – Research Integration and Formal Methods Workbench
Rockwell Collins and University of Minnesota


ARCHITECTURE-CENTRIC PROOF
Formal System Requirements
Control Components Verification and Synthesis
Contracts
Verified Components
System Architecture System Design Compositional Verification and Synthesis
Verified Vehicle Code
Open Source Vehicle
Military Vehicle
HACMS_002_SWEng_11



16 months into the project

Draper Labs could not hack into the system in 6 weeks

Had access to source code

Technical Approach

- Develop a complete, formal architecture model for UAVs that provides robustness against cyber attack
- Develop compositional verification tools driven from the architecture model for combining formal evidence from multiple sources, components, and subsystems
- Develop synthesis tools to generate flight software for UAVs directly from the architecture model, verified components, and verified operation system

Accomplishments

- Created AADL model of vehicle hardware & software architecture
- Identified system-level requirements to be verified based on input from Red Team evaluations
- Developed Resolute analysis tool for capturing and evaluating assurance case arguments linked to AADL model
- Developed example assurance cases for two security requirements
- Developed synthesis tool for auto-generation of configuration data and glue code for OS and platform hardware

Open source tools available at github.com/smaccm

Agenda

Challenges in Software Reliant Systems

Four Pillar Improvement Strategy

Virtual System Integration

Incremental Lifecycle Assurance

Benefits of Virtual System Integration & Incremental Lifecycle Assurance

Reduce risks

- Understand system wide impact early
- Verify assumptions across system

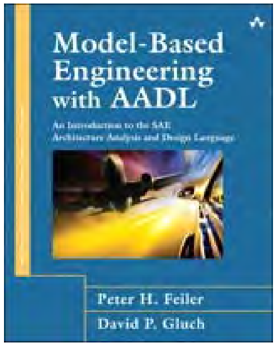
Increase confidence

- Verified models to complement integration tests
- System design evolved from verified models

Reduce cost

- Fewer system integration problems
- Less assurance related rework

References


Model-Based Engineering with AADL
An Introduction to the SAE Architecture Analysis and Design Language
Peter H. Feiler
David P. Gluch

AADL Website www.aadl.info and AADL Wiki www.aadl.info/wiki

Blog entries and podcasts on AADL at www.sei.cmu.edu

AADL Book in SEI Series of Addison-Wesley
http://www.informit.com/store/product.aspx?isbn=0321888944

On AADL and Model-based Engineering

http://www.sei.cmu.edu/library/assets/ResearchandTechnology_AADLandMBE.pdf

On an architecture-centric virtual integration practice and SAVI

http://www.sei.cmu.edu/architecture/research/model-based-engineering/virtual_system_integration.cfm

On an a four pillar improvement strategy for software system verification and qualification

http://blog.sei.cmu.edu/post.cfm/improving-safety-critical-systems-with-a-reliability-validation-improvement-framework

Webinars on system verification https://www.csiac.org/event/architecture-centric-virtual-integration-strategy-safety-critical-system-verification and on architecture trade studies with AADL https://www.webcaster4.com/Webcast/Page/139/5357

Contact Information

Peter H. Feiler
Principal Researcher
RTSS
Telephone: +1 412-268-7790
Email: phf@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web
Wiki.sei.cmu.edu/aadl
www.aadl.info

Customer Relations
Email: info@sei.cmu.edu
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257