

Incremental Lifecycle Assurance of Critical Systems

Peter Feiler



Copyright 2016 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.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

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.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

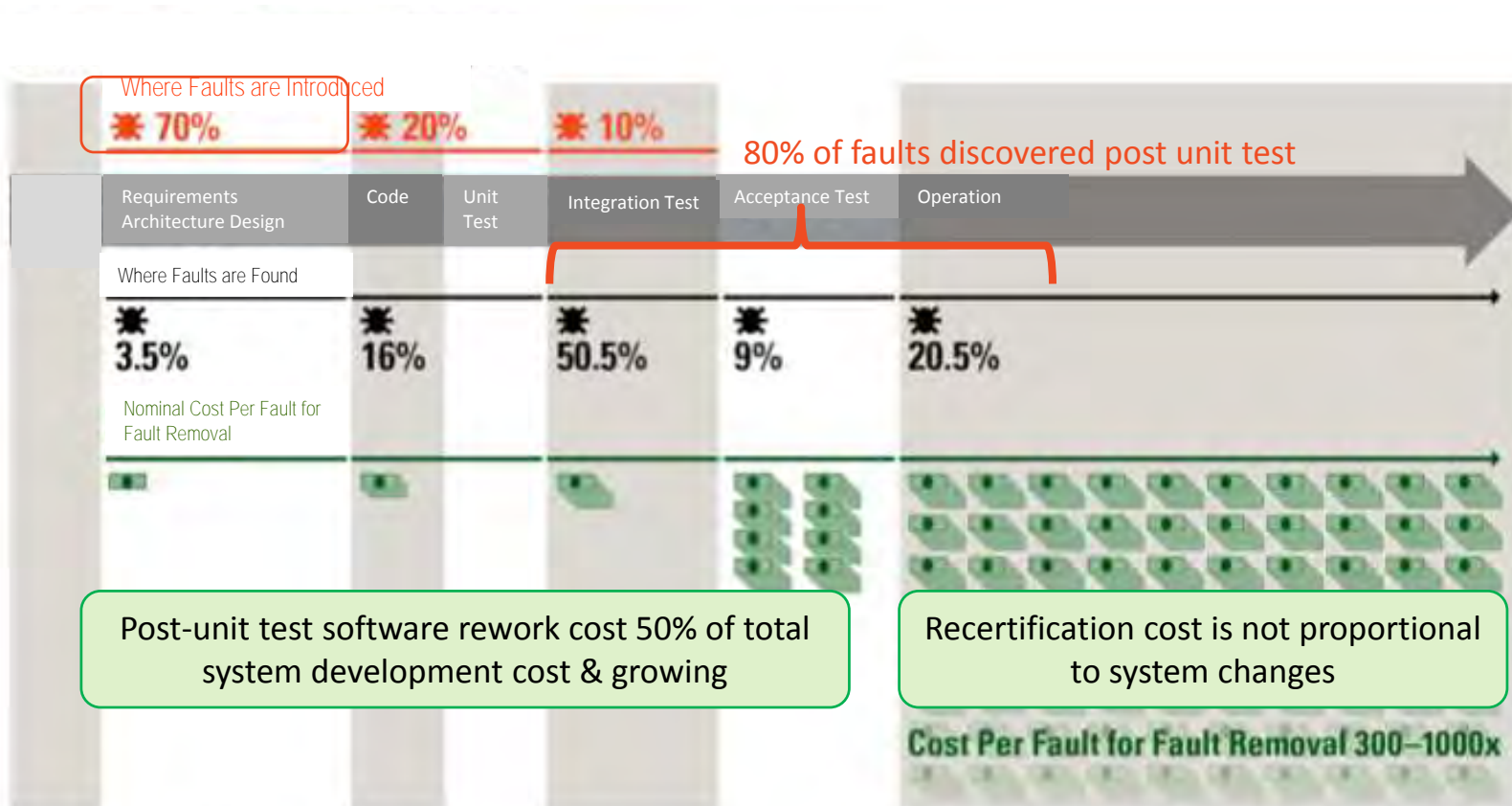
DM-0004087

Outline



Critical System Assurance Challenges **Incremental Lifecycle Assurance Approach** **ALISA Workbench**

Critical System Assurance Challenges



Sources: Critical Code; NIST, NASA, INCOSE, and Aircraft Industry Studies

Years between labor-intensive system safety assessments
 Software as major hazard source often ignored

Requirements and 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.

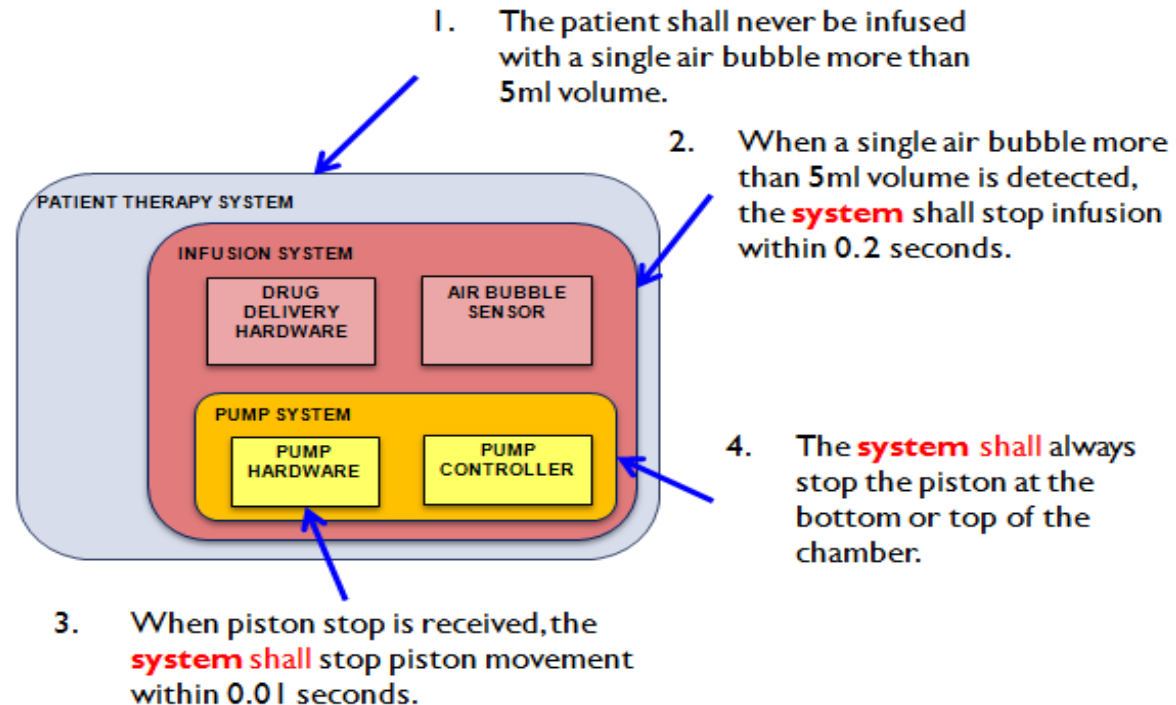
When piston stop is received, the **system shall** stop piston movement within 0.01 seconds.

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

U Minnesota Study

Importance of understanding system boundary

Same Requirements Mapped to an Architecture Model



NIST Study

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

We have effectively specified a system partial architecture

Outline



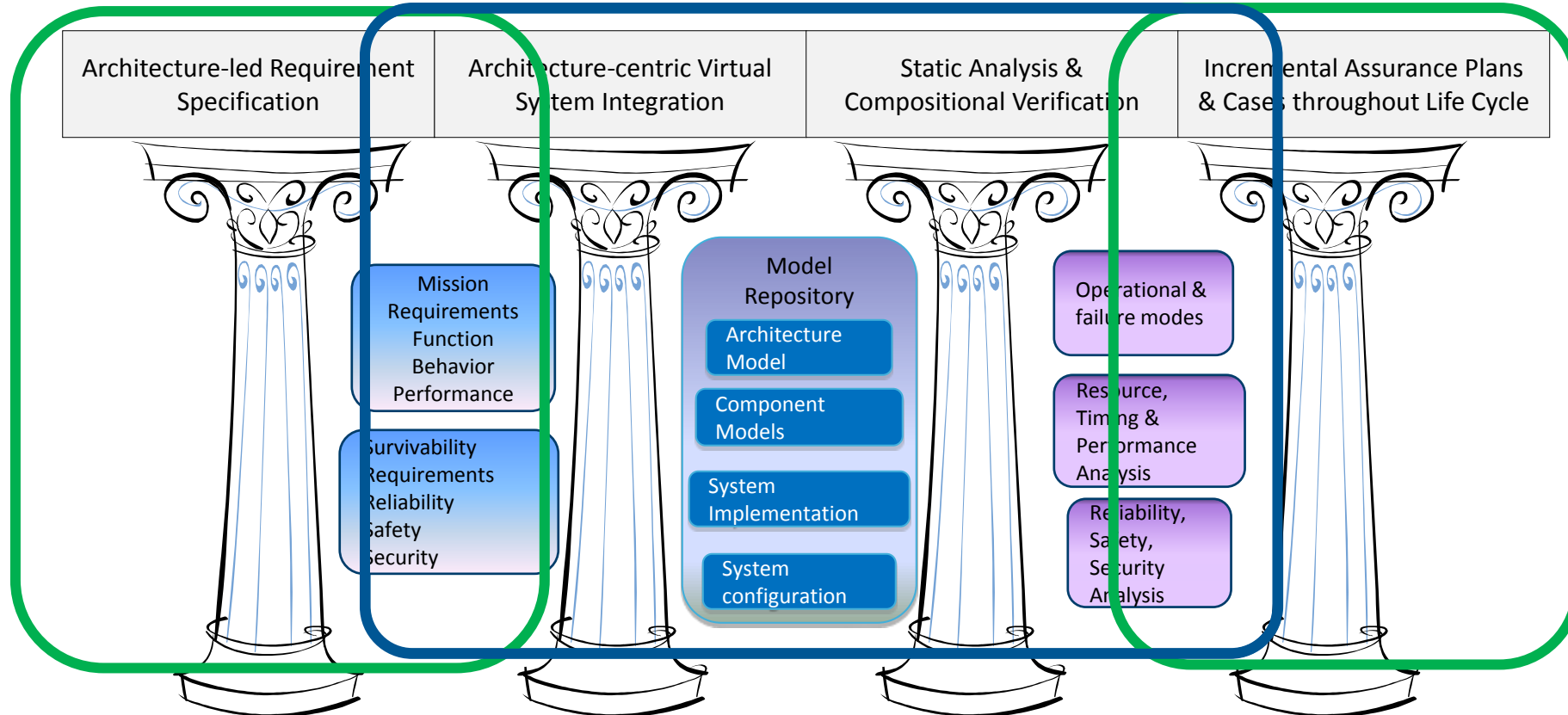
Critical System Assurance Challenges

Incremental Lifecycle Assurance Approach

ALISA Workbench

Assurance and Qualification Improvement Strategy

Assurance: Sufficient evidence that a system implementation meets system requirements



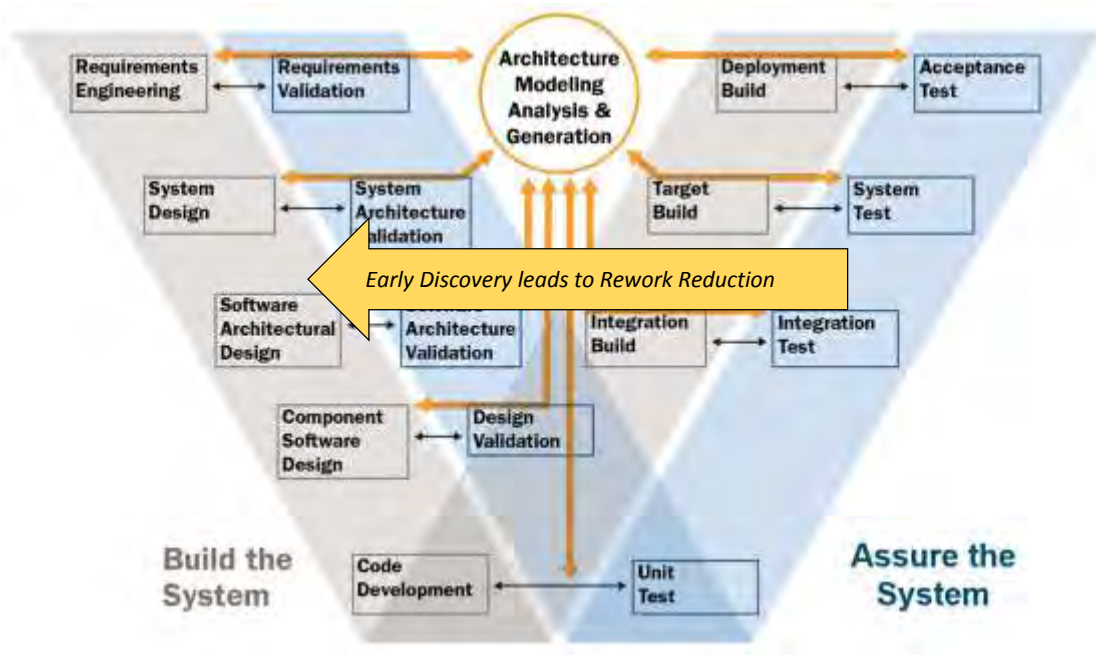
Early Problem Discovery through Virtual System Integration and Analysis
Improved Assurance through Better Requirements and Automated Verification

2010 SEI Study for AMRDEC
Aviation Engineering Directorate



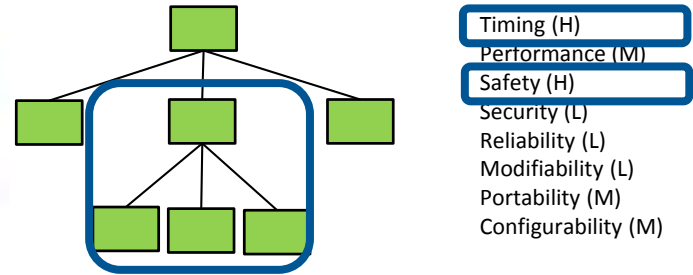
Three Dimensions of Incremental Assurance

Incremental assurance throughout lifecycle
 Early discovery through virtual system integration
 Return on Investment study by SAVI*

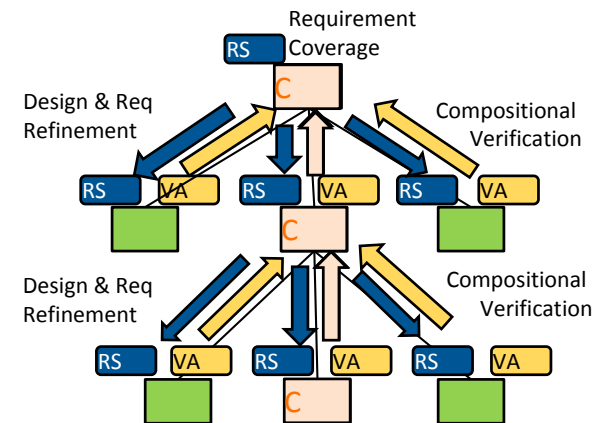


*System Architecture Virtual Integration (SAVI) Aerospace industry initiative

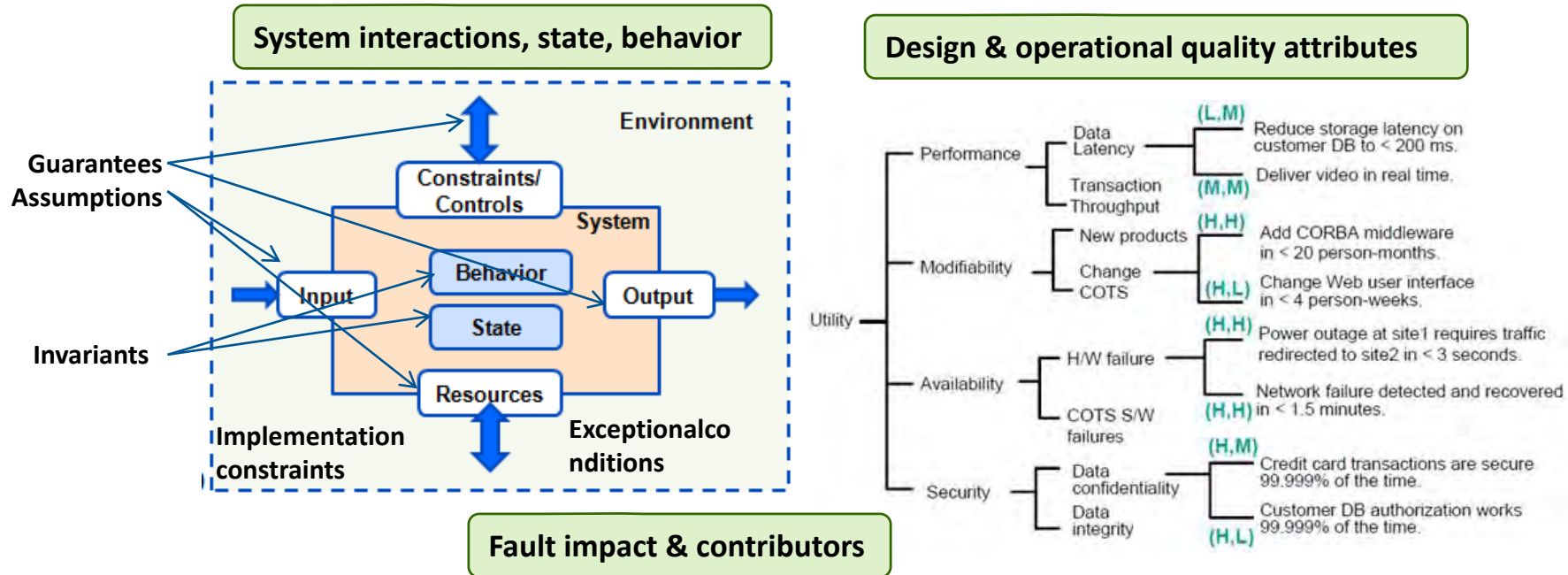
Priority focused architecture design exploration for high payoff
 Measurable improvement (Rolls Royce)



Compositional verification and partitions to limit assurance impact

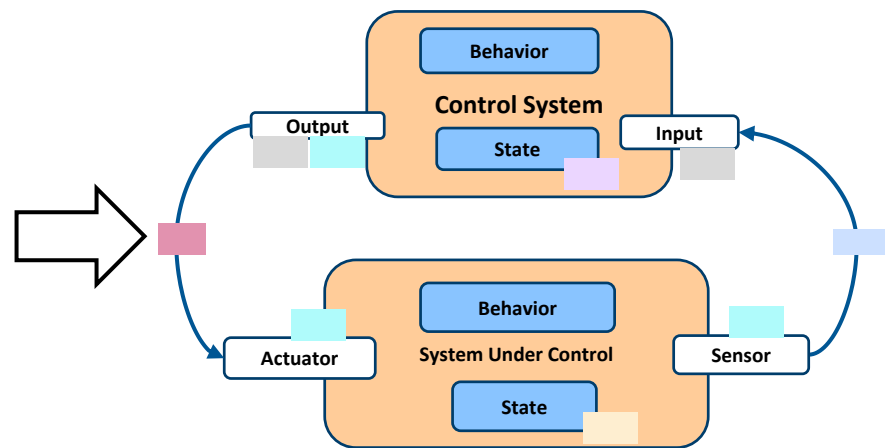


Three Dimensions of Requirement Coverage



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

Fault Propagation Taxonomy





Impact and Alignment

DoD Acquisition and Industry Organizations

- OASD R&E: Champion maturation and insertion of virtual system integration into DoD programs
- DARPA research successes in HACMS program
- AMRDEC Joint Multi-Role (JMR) Tech Demo: maturation of Virtual System Integration for Future Vertical Lift (FVL) program
- Aerospace industry System Architecture Virtual Integration (SAVI) initiative Multi-year investment: Boeing, Airbus, Embraer, suppliers, FAA, NASA, DoD
- Rolls Royce engine control system case study

Standard Development

- Draft SAE AADL Requirement Specification standard
- Revision of SAE S18 ARP4761 System Safety Analysis standard

Regulatory Certification Agencies

- FDA: Guidance on medical device (re-)certification
- Underwriters Lab: medical device integration guidance (AAMI/UL2800)
- NRC: Educational workshop series on software system assurance

Outline



Critical System Assurance Challenges **Incremental Lifecycle Assurance Approach** **ALISA Workbench**



Modeling Notations in ALISA Prototype

ReqSpec Represent stakeholder and system requirements

- Document-based and architecture-led
- Verifiable system requirements
- Coverage and uncertainty

Verify Specify intended verification activities

- Across lifecycle on different artifacts and layers of system architecture
- Via verification methods (manual, automated)
- Supported: OSATE Analyses, Java, Resolute, Agree, JUnit

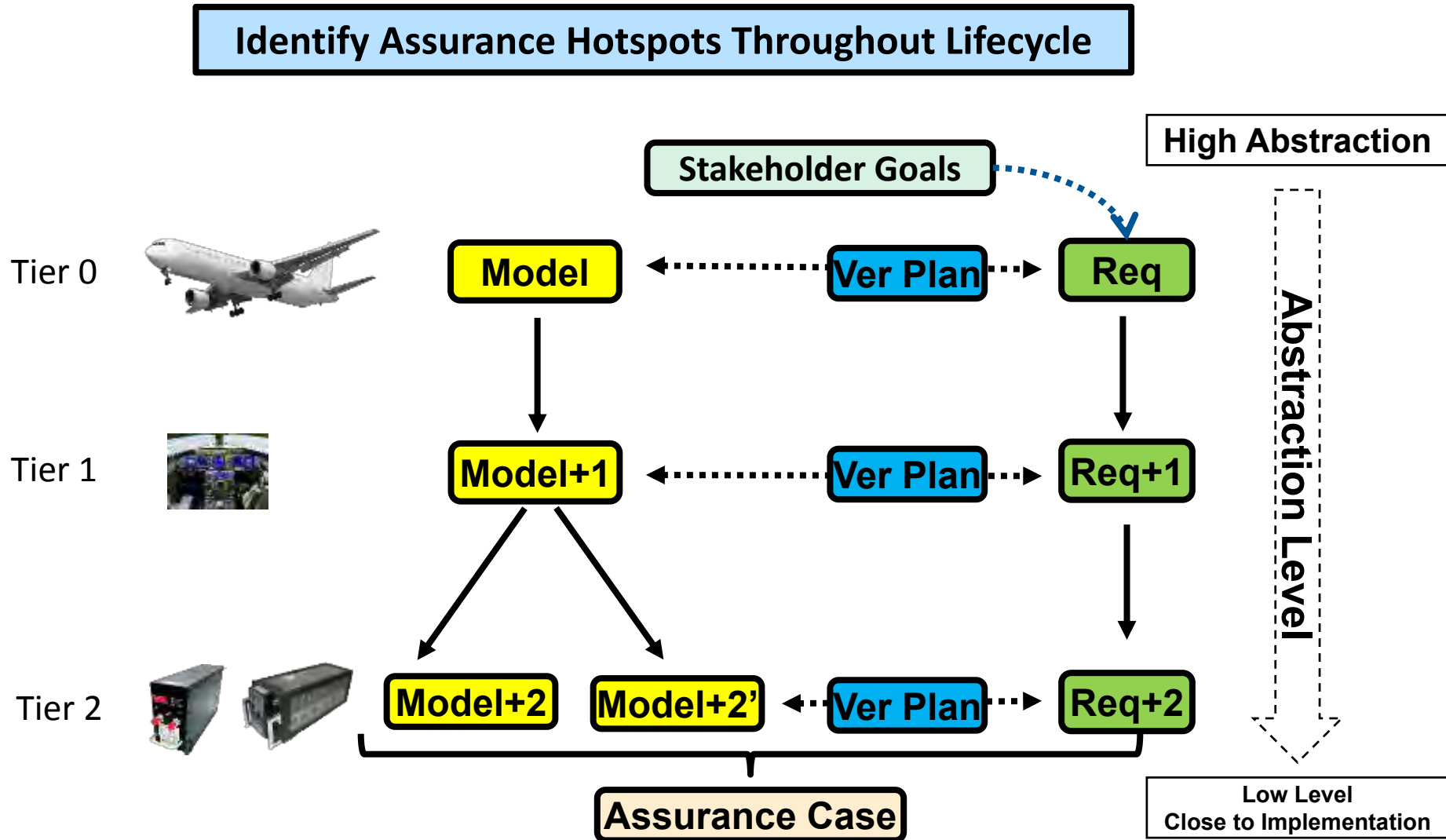
Alisa Compositionally configure assurance cases

- Reasoning logic of how verification activities satisfy requirement
- Assumptions, preconditions on verification activities
- Scoped assurance plans and focused assurance tasks

Assure Manage assurance state and results

- Multi-valued logic evaluation of verification action and results
- Acceptable risk factors (e.g., design assurance levels)
- Time phased execution of assurance plans

Automated Incremental Assurance Workbench



ALISA Workbench Screenshot

The screenshot displays the ALISA Workbench interface for the project 'AADL Alisa-IntegratorDemo/alisa/Aircraft.alisa - OSATE2'. The main editor shows AADL code for an assurance case named 'SAVI' for 'AircraftSystem::AircraftSystem'. The code includes two assurance plans: 'SaviDemo' and 'AircraftTier2', both of which assure the 'AircraftPlan' and its subsystems 'FGS ENG' and 'HYD APU'. An assurance task 'Tier2SafetyNetworkFocus' is also defined with a category of 'Quality.NetworkUtilization Quality.Safety'.

The Assurance Results table shows the following data:

Evidence	0	5	1	Description	results count
Assurance case SAVI	[Progress bar]				(\$180 F4 T0 E0 tbd0 E)
Assurance plan SAVI.SaviDemo	[Progress bar]				(\$10 F2 T0 E0 tbd0 EL)
Claim R1(AircraftSystem)	[Progress bar]			The weight of the Aircraft system...	(\$3 F0 T0 E0 tbd0 ELO)
Subsystem verification FGS	[Progress bar]				(\$6 F0 T0 E0 tbd0 ELO)
Subsystem verification ENG	[Progress bar]				(\$0 F1 T0 E0 tbd0 ELO)
Claim R1(Engines)	[Progress bar]			The weight of the Engine shall n...	(\$0 F1 T0 E0 tbd0 ELO)
Evidence weightlimit	[Progress bar]			Perform full weight (mass) analys...	(\$0 F1 T0 E0 tbd0 ELO)
Subsystem verification ELE	[Progress bar]				(\$1 F1 T0 E0 tbd0 ELO)
Assurance plan SAVI.AircraftTier2	[Progress bar]				(\$170 F2 T0 E0 tbd0 E)
Claim R1(AircraftSystem)	[Progress bar]			The weight of the Aircraft system...	(\$3 F0 T0 E0 tbd0 ELO)
Subsystem verification ENG	[Progress bar]				(\$0 F1 T0 E0 tbd0 ELO)
Subsystem verification ELE	[Progress bar]				(\$1 F1 T0 E0 tbd0 ELO)
Subsystem verification FGS	[Progress bar]				(\$166 F0 T0 E0 tbd0 E)

The Alisa View panel on the right shows a 'Select Filter:' dropdown menu with 'None' selected. Below the dropdown are two buttons: 'Verify all in Assurance Case with Selected Filter' and 'Verify remaining in Assurance Case with Selected Filter'.



Assurance Case Execution and Metrics

User guided filtered views

- Filtering on requirement type, quality attribute, development phase
- User definable categories for requirements, verification methods and activities

Assurance Metrics

- Requirement coverage measures
 - Model element, quality, and failure effect taxonomy coverage
- Multi-valued verification result measures and their aggregates
 - Pass, fail, incomplete, conditional, backups
- Weighted requirement claims, verification activity results
 - Reflect importance, uncertainty (volatility, precedence, impact)

Guidance throughout lifecycle (Spotlight)

- Based on requirement specifications and precedent and volatility ratings
- Utilize COCMO II to derive worst-case and best-case estimates of effort

Case Studies

Multi-Tier Aircraft Model

- Demonstrate incremental and compositional approach to assurance cases

Stepper Motor diagnostics and design verification

- Demonstrate diagnostic of original customer design and verification of three design improvements

Situational awareness system

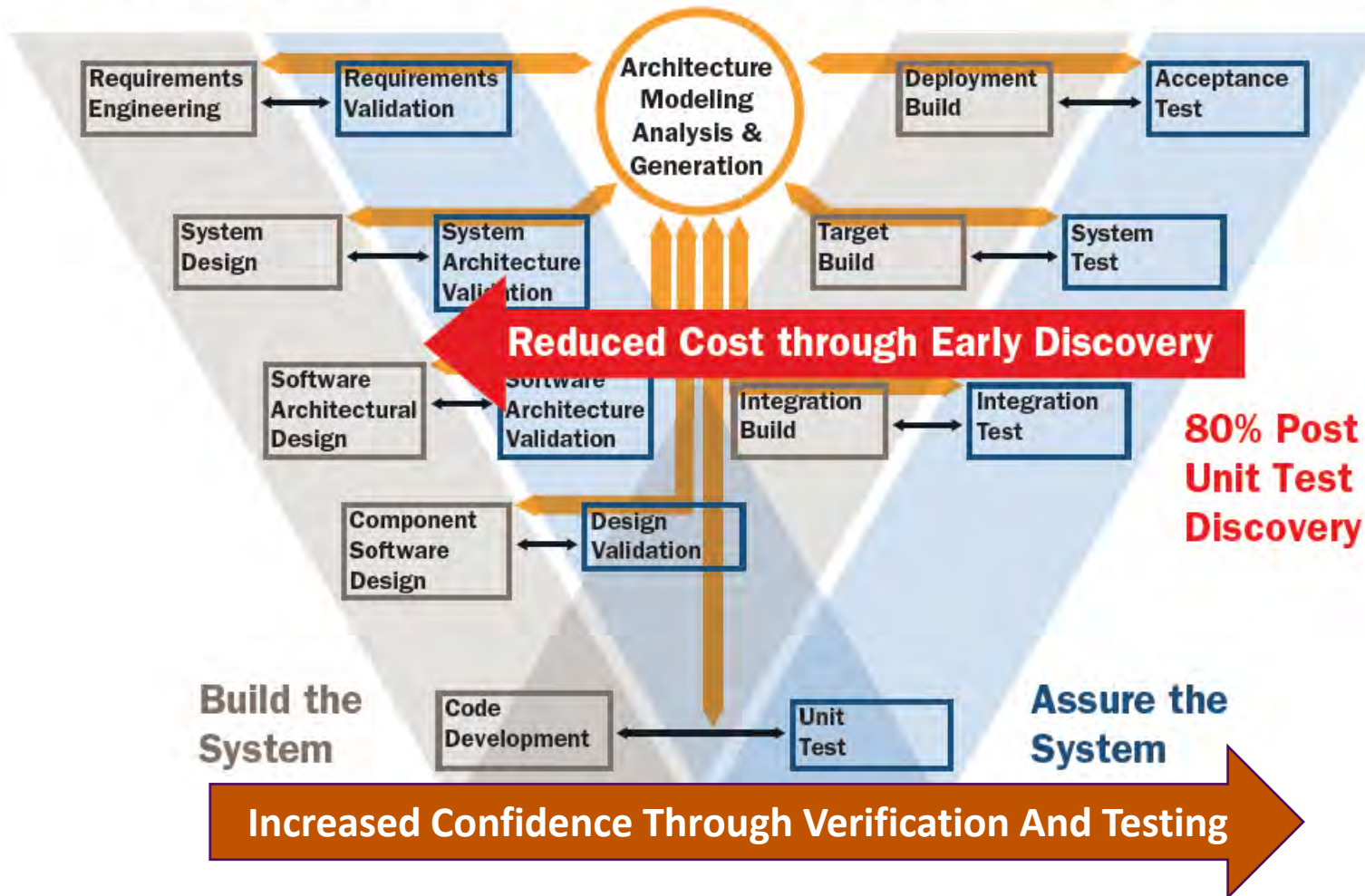
Requirement Spec & Virtual System Integration Case Study

- **Joint Common Architecture (JCA) Demo**
 - Model based acquisition of FACE conformant software
 - Integration onto multiple Operating Environments
- **Architecture-Centric Virtual Integration Process (ACVIP)**
 - Shadow Effort to JCA Demo after BAA was released
 - By Software Engineering Institute (SEI), Adventium Labs, Software Engineering Directorate (SED)
- **Discovered potential system integration issues in advance through requirements, safety and timing analyses**
 - Early identification of 85+ potential integration issues

Architecture analysis is critical for the successful and affordable integration of systems!

20

Benefits of Virtual System Integration and Incremental Lifecycle Assurance



Contact Information

Presenter / Point of Contact

Peter H. Feiler

SEI Fellow

Telephone: +1 412.268.7790

Email: phf@sei.cmu.edu