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.


DM-0004087
Outline

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

- **Nominal Cost Per Fault for Fault Removal**
  - Requirements: 3.5%
  - Architecture Design: 15%
  - Code: 50.5%
  - Unit Test: 9%
  - Integration Test: 20.5%
  - Acceptance Test: 80%
  - Operation: 80%

- Post-unit test software rework cost 50% of total system development cost & growing
- Recertification cost is not proportional to system changes
- Years between labor-intensive system safety assessments
- Software as major hazard source often ignored

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

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.

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

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

We have effectively specified a system partial architecture

NIST Study

<table>
<thead>
<tr>
<th>Requirements error</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete</td>
<td>21%</td>
</tr>
<tr>
<td>Missing</td>
<td>33%</td>
</tr>
<tr>
<td>Incorrect</td>
<td>24%</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>6%</td>
</tr>
<tr>
<td>Inconsistent</td>
<td>5%</td>
</tr>
</tbody>
</table>
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

---

Incremental Lifecycle Assurance of Critical Systems

Oct 2016
© 2016 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.
Three Dimensions of Incremental Assurance

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

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

Compositional verification and partitions to limit assurance impact

*System Architecture Virtual Integration (SAVI) Aerospace industry initiative
Three Dimensions of Requirement Coverage

- System interactions, state, behavior
  - Guaranteed Assumptions
  - Invariants
  - Implementation constraints

- Design & operational quality attributes
  - Constraints/Controls
  - System
  - Behavior
  - Output
  - Resources
  - Exceptional conditions

Three Dimensions of Requirement Coverage

- System interactions, state, behavior
- Design & operational quality attributes
- Fault impact & contributors

Fault Propagation Taxonomy

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

Stakeholder Goals

Identify Assurance Hotspots Throughout Lifecycle

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

Low Level Close to Implementation

High Abstraction

Model+2'
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
Benefits of Virtual System Integration and Incremental Lifecycle Assurance

Increased Confidence Through Verification And Testing

Build the System

Assure the System

Reduced Cost through Early Discovery

80% Post Unit Test Discovery

Incremental Lifecycle Assurance of Critical Systems

© 2016 Carnegie Mellon University
Contact Information

Presenter / Point of Contact
Peter H. Feiler
SEI Fellow
Telephone:  +1 412.268.7790
Email:  phf@sei.cmu.edu