Test Perspectives for Architecture

Joe Manas
Beth Wilson
Raytheon

NDIA Systems Engineering Conference
October 2014
Test Perspectives for Architecture

- **Strategic Partnerships**
  - Test strategy concepts
  - Architecture views with test strategy insight

- **Successful Applications**
  - Program #1: Architecture feedback during test planning
  - Program #2: Additional architecture products to support test planning
  - Program #3: Test team engagement with architecture development

- **Sample Conversations**
  - Define incremental capability
  - Partition functionality

Streamline Test Program While Reducing System Complexity
Strategic Partnerships
Need Test Planning to be Strategic

If you compared your integration effort to a soccer team, would it be the way 8 year olds play or the way professionals play?

How much does your verification success depend on SMEs and heroes?

Transition Luck and Heroics into Strategy
Integration and Verification Strategy Driven by System Architecture

Principles
- Minimize testing on the ship
- Minimize duplication
- Minimize product verification on the ship
- Early verification
- Test at the right time (minimize retest)
- Maximize reuse of procedures
- Leverage other verification activities

Integration and verification flows based on incremental capability

Strategy

Plan

Cases

Procedure

Storyboard

Conduct

Integration and verification complexity based on functional partitions

Activity flow that reflects application of principles

Activity planning that establishes handoffs between teams, activity constraints, and task content

Detailed description of planned actions and expected outcomes

Report of success and failure

Integration Case

Test Case

Title

Type

Objective

Prerequisite Tests

Supporting Test Equipment and Assets

Personnel Required

Estimated Duration

Method

Requirements Verified

Summary of verification event purpose

Automated number from HPQC

Summary of verification event approach

Subsystem and/or Combat Systems requirements

May be separate phases with different personnel

Conduct, Support, Witnesses, etc.

Resources, TLSF capability, etc.

Detailed Report of success and failure

Conduct and failure

11/3/2014 5
Useful Views for Test Strategy

Incremental OV-1s Operational Concept

SV-4 Systems Functionality Description

CV-3 Capability Phasing

OV-6c Event Trace Description

OV-5b Operational Activity Model

Test Can be a Stakeholder for Architecture
Successful Applications

Program #1: Architecture feedback during test planning
Program #2: Additional architecture products to support test planning
Program #3: Test team engagement with architecture development
Program #1
Complexity Feedback

Are there changes in the architecture that can simply integration (and design)?

Interface Ranking

Test Architect Can Simplify Interfaces
Program #2
Additional Architecture Products

Fishbone from Test Architect

OV-1s from System Architect

Resulting SV-8 Depicting Incremental (and Early) Capability

SV-8 System Evolution

Incremental capability as pursuit win theme
Program #3: IV&V Tactics for Quality Attributes

<table>
<thead>
<tr>
<th>Quality Attribute</th>
<th>HW</th>
<th>SW</th>
<th>IV&amp;V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interoperability</td>
<td>COTS standard hardware</td>
<td>Open architecture</td>
<td>Low risk interfaces</td>
</tr>
<tr>
<td>Usability</td>
<td>Reduce number of monitors</td>
<td>Intuitive workflow processing</td>
<td>Early integration of HMI to influence user acceptance</td>
</tr>
<tr>
<td>Scalability</td>
<td>Computing environment with growth</td>
<td>Multi-threading</td>
<td>Focus on integration strategy and capability build up</td>
</tr>
</tbody>
</table>
Conversations

Define incremental capability
Partition functionality
Conversation #1: Capability Build-up

Test Architect

I need to define the integration strategy. Do you have a view of the architecture that shows how the capabilities come together?

System Architect

REAP doesn’t require that I do a Capability View, so I didn’t do that one.

It’s an important view for me. I have these test events, but don’t know how capability needs to be integrated to support them.

We can work together to define the incremental capability build-up. I think you are looking for a systems evolution description (SV-8), not a capability phasing (CV-3).
Conversation #1: Capability Build-up

I also need those lightning bolt charts you do for each event.

I only have the complete system view. We can develop OV-1s to reflect each node.

Sub-system A won’t be available at this event, but I can build an emulator for that. I can’t do that for sub-system B – is there any way to change the architecture to allow a more incremental build-up?

I could change the functional allocation. The customer really wants early capability so that would work well.
Conversation #2: Partitioning Functionality

Test Architect

Here is the problem. Is there any way to simplify this?

System Architect

I can partition the functionality differently. Does this help?

Much better!

This interface is ranked as a high-risk integration effort. Why is it so complicated?

Here is the SV-4 (Systems Functionality Description) for that.
Summary

- Test and Architecture interaction is a rich opportunity
  - Design streamlined test program
  - Reducing system complexity.

- Test and Architecture partnerships have been successful
  - Incremental capability
  - Simplified interfaces
  - Test impact on quality attributes

Look for Test Perspectives for Architecture Products
Abstract

The intersection between test strategy and architecture development is rich with opportunity to design a streamlined test program while reducing system complexity. The presentation will highlight three successful applications of this overlap showing that the test strategy can provide feedback to simplify the architecture and architecture products can be defined that will define the test strategy. The presentation will then identify test perspectives for architecture development through hypothetical conversations between the test and architecture leads. The examples focus on defining incremental capability and partitioning functionality.
Biographies

- **Joe Manas** is an Engineering Fellow with Raytheon Company. Over the last 28 years, he has worked within the defense & aerospace industry, 25 years of which has been with Raytheon. Joe has held leadership positions in the disciplines of System Engineering, Software Development and Test & Evaluation across multiple product lines. He holds a B.S. in Electrical Engineering from Worcester Polytechnic Institute, MA.

- **Dr. Beth Wilson** is a Senior Principal Engineering Fellow who earned her PhD in Electrical Engineering from the University of Rhode Island. Since joining Raytheon in 1983, she has worked as a design engineer, program manager, research scientist, functional manager, and test director on sonar, satellite, and radar programs. She is the NDIA Developmental Test and Evaluation committee chair, and co-chair for the INCOSE and NDIA System Security Engineering committees. Previous assignments have included a character-building deployment to Shemya, Alaska as the Test Director for the Cobra Dane Upgrade. Beth is a Raytheon Certified Architect (RCA), INCOSE Expert Systems Engineering Professional (ESEP), and a Raytheon Certified Six Sigma Expert.