Assurance Cases and Test Design Analysis

Flight Software Workshop
November 7-9, 2012

Khalid Lateef Ph.D.
Outline

• Scope
• Test Design challenges
• Assurance cases
• Example of test design IV&V for an automobile
• Triggers – How to find the right triggers
• Test Scenarios – All scenarios not created equal
• Results from 2 CSCIs
• Summary and Conclusions
Scope and Acknowledgement

• Scope
  – Small part of the much larger risk assessment study
  – Work initiated last year
  – Validating Assurance cases approach against the Test analysis work already performed using traditional approach (CoM)

• Acknowledgment
  – IV&V supported Assurance case assessment (2011)
  – Other NASA centers
  – Non-NASA groups (SEI-CMU, Aerospace Corp, Adelard UK)
Test design V&V

- Test Results Verification
  - Test design to wring out the bugs
  - Was this effort successful?
- Test design Validation – Nominal behaviors
  - Relatively straightforward
  - Not many issues discovered
- Test Design Validation – Off nominal behaviors
  - Takes more thought. What can go wrong? What shouldn’t it do? Off nominal behavior. Good number of issues
  - What is “appropriately”? Off nominal behavior. Ripe for issues/finding bugs
  - Application the Safety Critical / Space Systems
Off-nominal test design

DO-178B

• Normal Range Test Cases:
  – Boundary values on input variables
  – Multiple iterations for time-related functions
  – Transitions for state based software

• Robustness Test Cases:
  – Invalid values for variables
  – System initialization under abnormal conditions
  – Failure modes of incoming data
  – Exceeded time frames
  – Try to provoke illegal state transitions
  – Arithmetic Overflow
  – Loop counts
Possible Inputs for the Test Design Analysis

• Validated SW requirements
• Test artifacts
  – Test Plan
  – Test procedures
  – Test scripts
  – Test Logs
• Test artifacts associated with multiple builds
• Con Ops, User manuals, Interface documents
• Test validation scope based on PBRA and RBA
Testing challenges-Space System

• System Initialization
  – Timing constraint
  – Init Failure?
    • Response from other systems or ground

• Startup image management
  – Auto switch to backup image?
    • Appropriate bits commandable?
Testing challenges-Space System (Contd.)

• System Safety
  – Fault Detection
    • Fault levels (1, 2, or low level 3 fault)
  – Fault response
    • Autonomous/Manual Response enabled/inhibited
    • Abort sequences (if applicable)
    • Commands to enable / disable response, reset flags
    • Swapping strings (IMOK monitoring)
  – Preventative measures
    • Arm/fire commands
    • Command processing (FSW validates? Executes?)
Assurance Case

• What is an assurance case?
  – Specialized instance of general case argumentation\(^1\)
    • Claim $\leftarrow$ Evidence (Build an argument using Evidence for a given claim)
  – Claims can have sub-claims

• Tools
  – ASCE
  – Excel
Example: Test design analysis

• Test design
  – From the Test team
  – Before the car is ready for full scale production or
  – A batch of cars is ready to be shipped to the dealer / customer
Claims

• Claim#1: Radio/MP3 Player will work
• Claim#2: Dome light will work
• Claim#3: Test driver can drive the car along an intended course
  – Sub-Claim#3.1: Car will start
  – Sub-Claim#3.2: Car will stop
Claim score based on safety, criticality

- Score 1..5 each of the sub-claims
  1. Engine starts up (safely)
  2. Car can be stopped safely
  3. Airbag will deploy in case of an accident

- Prioritization of claims based on score
Engine Startup claim-evidence analysis

• Startup scenario
  – Key in ignition → Turn clockwise → xx seconds → Engine started

• Claim: Engine will startup safely
  – Evidence: Ignition control will not trigger if Shift selector is in drive (forward or reverse)
  – Evidence: Ignition control will only trigger if the break pedal is in the specified position
  – Evidence: Ignition control will not trigger if the engine already running
  – Evidence: Ignition control will not trigger if the fuel pump or battery constraints are violated

Complexity of Test

Software testing is not about proving conclusively that the software is free from any defects, or even about discovering all the defects. Such a mission for a test team is truly impossible to achieve. Rex Black, Pragmatic Software Testing, John Wiley & Sons 2007
## Space System analysis

<table>
<thead>
<tr>
<th>Claim Level 1</th>
<th>Claim Level 2</th>
<th>Claim Level 3</th>
<th>Claim Level 4</th>
<th>Claim Level 5</th>
<th>Needed Evidence</th>
<th>Actual Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>Score</td>
<td>Score</td>
<td>Score</td>
<td>Score</td>
<td>Weight</td>
<td>Requirement</td>
</tr>
</tbody>
</table>

---

14
Test scenario

Trigger -> Response

• For each Claim, generate the list of triggers
• For each trigger generate test scenarios
• Test scenario
  – The trigger for the requirement(s),
  – Corresponding requirements and
  – The type of data being processed / touched by the requirement(s)
Trigger -> Response

Note 1: Annotations were added for the reference documents

Note 2: Generic diagram/table in the backup slides
Space System Triggers / Responses

• Triggers
  – External commands / HW telem aka across the interfaces
  – Internal (a relatively small number) to the system
• Group the triggers (Single / multiple interfaces)
  • User cmd impacting user interface only
  • User cmd impacting User interface and hw interface
• Responses
  – Internal to the system
  – To the external interfaces
Test Design Validation Analysis & Evidence

– Test Scenario
  • Test scenario trigger
  • Test scenario step #
  • Step description / behavior

– Reference info
  • Source (document section number, Req tag number)
  • Safety or criticality related to the test step
  • Adverse conditions (if any)

– Evidence info
  • Correlation to the test plan section
  • Correlation to the test procedure (number, step)
  • Correlation to the test script (code line number)

– Observations / Issues (if any)
Test Design Issues verified

- Incomplete Arm / fire Commands tests
- Missing “Alternative” steps in the abort scenario tests
  - Off nominal for abort-sequence
- Inadequate fault flag responses tests
- Incomplete Command parameter verification tests
- Missing mode verification tests
Two CSCIs of a Space system

- ~ 250 requirements each (Validated)
- ~ 45 ground commands each
- Ground/SW interface
- SW/HW interface
- ~60 test scripts each
  - One with separate test design
  - The second with high-level test procedure embedded in the test script (as comments)
Summary

Assurance cases can help to

• Develop comprehensive test scenarios
• Systematic steps to uncover Off nominal conditions
  – Off nominal conditions are the source of high severity issues with Test design and the system being tested
• Identify and use system triggers as part of the test design analysis
• Look for safety-critical test scenarios
• Verify the test results
• Review the issue resolutions for additional/new bugs
Future Work

Using Assurance cases for

• Analyzing test design
• Test Coverage assessment
  – Automated mapping
• Independent testing
Questions ?
Backup slides
Verifying the test results

• Test results Review
  – Test logs
  – Test terminal screen dumps

• Test results show
  – Commands executed
  – Triggers identifiable
  – Trigger occurred at the correct time
  – System responses as expected
  – Time stamps show if any deadlines violated