Comparison of IV&V’s Use of Non Verification Environments and Program Use of Verification Environments

2014 Annual Workshop on Verification and Validation
September 9-11, 2014

Presenter:
Ricky Beamer
frederick.m.beamer@ivv.nasa.gov

Additional Author:
Katie Warner
kcwarner@mix.wvu.edu
• IEEE-1012 Review
• Dynamic Environment Retest
• MPCV Overview
• Test Environment Overview
• Testing Methodology
• Process
• Results
• Forward Work
IEEE “Standard for System and Software Verification and Validation”

- Determine whether the development products conform to the requirements satisfies its intended use
- Verification and Validation processes include Systems, Software, Hardware and Interfaces
- Standard applies to systems, software, and hardware being developed, maintained, or reused
- Verification and Validation processes include the analysis, evaluation, review, inspection, assessment, and testing of products
IEEE 1012 Appendix C

For system tools, technical independence means that the IV&V effort uses or develops its own set of test and analysis tools separate from the developer’s tools. **Sharing of tools is allowable** for computer support environments (e.g., compilers, assemblers, and utilities) or for system simulations where an independent version would be too costly. For shared tools, IV&V conducts qualification tests on tools to assure that the common tools do not contain errors that may mask errors in the system being analyzed and tested. Off-the-shelf tools that have extensive history of use do not require qualification testing. The most important aspect for the use of these tools is to verify the input data used.
Due to cost and schedule limitations and the nature of the EFT-1 flight test, conducting full qualification tests on the provided MPCV test environments was not pursued.

- IV&V MPCV executed supplied checkout and regression scripts to ensure that the installed simulators behave in a similar manner as the Program counterpart. This ensures that both systems behave in the same way for the same broad range of test scenarios.
- This strategy has been used on other IV&V projects when full qualification tests were not practical.
Dynamic Environment Retest

- Dynamic Testing: a method of assessing the feasibility of a software program by giving input and examining output.
- The purpose of IV&V’s retest of the Program supplied test environments was:
  - Determine if test environments display anomalies which merit funding additional IV&V tests.
  - Establish environments’ utility for future planned IV&V testing.
  - Establish greater confidence in previous IV&V test results.
    - Ensure observed behaviors of software under test (SUT) are due to SUT performance not due to test environment performance.
  - Begin process for future independent qualification of the environments.
MPCV Overview

- Multi-Purpose Crew Vehicle (MPCV) is based on the Constellation program’s Orion design
- Capable of conducting regular in-space operations in conjunction with payloads delivered by the Space Launch System (SLS) for missions beyond LEO
• Expected to launch in December 2014
Test Environment Overview

• SOCRRATES: Software Only Crew Exploration Vehicle Risk Reduction Analysis Test Environment Simulation

• Developed by Lockheed Martin (LM) for the purpose of MPCV Flight Software (FSW) development testing and verification procedure development (verification test dry runs)

• SOCRRATES has three main components
  – SOCRRATES Controller
  – MPCV FSW
  – OrionSim

• Emulates execution of the FSW on the Vehicle Management Computer (VMC) through Wind River Simics software only hardware models
PLATO: Partition Level Application Test for Orion

- Software-based execution environment developed by NASA at the John Space Center that allows users to interact with the Orion FSW partitions at their sampling port boundaries
  - Sampling ports are shared memory within the partitioned environment used to pass information from ARINC 653 application partitions to other SW partitions or non partitioned software
  - MPCV uses for sampling ports include communication of commands, statuses, and responses via software that is assigned ownership of the communications bus

- Simulation environment that executes the unmodified MPCV FSW on a Simics emulated target
  - Allows for stepping through the FSW minor frames and controlling interpartition data transaction timing
Test Environment Overview

- Verification Test Environments
  - VTB – VMC Test Bench
  - ITL – Integration Test Laboratory
  - ATLO – Assembly, Test and Launch Operations

- Certified testing facilities
  - Allow for vehicle hardware and software testing

- Risk identification and mitigation facilities
Testing Methodology

- Test case selection
  - Examine available test case and test procedure documentation
  - Examine available RFR data
  - Correlate test case and RFR data
  - Prioritize test cases

- Execute test cases
  - Modify test scripts as necessary for the testing environment

- Examine and compare test data
  - Note differences and similarities between RFR and IV&V test data
• Re-executed the RFR test scripts
• All RFR data we were looking at was from several older versions of the FSW
• From this we knew that certain things, such as telemetry packet numbers, would change from build to build (observed from prior regression testing)
• Excluding that info, we looked to see if the data captured in the official RFR results was similar to our test results
  – Margin of error due to updated FSW and updated component models being used
### Results

#### PLATO

- Consistencies: 25
- FSW Update Deviations: 7
- Other Deviations: 0
- Total: 32

#### SOCRRATES

- Consistencies: 10
- Other Deviations: 2
- Total: 12

### Table

<table>
<thead>
<tr>
<th>Results</th>
<th>Consistencies</th>
<th>FSW Update Deviations</th>
<th>Other Deviations</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLATO</td>
<td>25</td>
<td>7</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>SOCRRATES</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td>12</td>
</tr>
</tbody>
</table>
Results

• Deviations
  – Due to more mature FSW being tested
    • In PLATO the FSW has changed to reflect different partition restart procedures. The test scripts have not been updated to reflect the altered logic
  – Development test system is unable to simulate all systems of systems
    • i.e. in SOCRRATES the component simulation system lacked the fidelity for the components under test

• IV&V’s use of the PLATO and SOCRRATES testing environments will support meaningful FSW testing of IV&V test cases without depending on the MPCV Program for test execution or test analysis
  – These environments are not demonstrated to be suitable final testing for the EFT-1 mission and confirmation of results via other environments is presumed
Forward Work

• Additional analysis of testing results

• Apply similar methodology to compare EFT-1 flight data and PLATO/SOCRRATES.
  – Establish OrionSim/PLATO/SOCRRATES representation of vehicle performance and software performance

• Examine the root cause for anomalies in flight data
  – Research if anomalous conditions were previously tested
  – Design test cases to recreate the anomalous condition
    • Test the test environment
    • Could the anomaly have been detected and mitigated prior to launch through dynamic testing?