Early Application of Computer Program Systems Integration, Test and Performance Measurement

Tom Sobieralski
Computer Sciences Corporation

NDIA National Test and Evaluation Conference
March 12-15, 2007
Agenda

• Background

• Systems Integration, Test and Performance Measurement

• Summary
Background
What is Computer Program Systems Integration, Test and Performance Measurement?

• The ability to verify the interfaces, functions and measure performance of two or more computer programs on the target hardware suite and operating environment.
Catalyst for Early Computer Program Systems Integration, Test and Performance Measurement

- Introduction of COTS hardware
- Conversion of legacy software to new languages
- System complexity with multiple hardware and operating environments
Multi-Processor Environments

• Legacy
  – AN/UYKs

• COTS Processors
  – Single Board Computers (SBCs)
  – Symmetric Multi-Processors (SMPs)
Multi-Operating Environments

• Legacy
  – Aegis Tactical Executive System (ATES)

• COTS
  – Concurrent Powermax
  – Sun Solaris
  – Red Hawk Linux
  – LynxOS
Multi-Computer Programming Languages

• Legacy
  – Compiler Monitor System -2 (CMS-2)

• New
  – Ada
  – C/C++
  – Java
Systems Integration, Test and Performance Measurement
Process Improvement

• Incremental Systems Integration, Test and Performance Measurement during the software development process
  – Improve System Stability
  – Early Identification and Resolution of issues and defects
  – Significantly reduce Engineering Test & Evaluation failure rates
  – Processes and Procedures QA reviewed and approved

CSC DMEI DE CMMI Level 5
Software Development Process

Development Phases
- System Requirements Analysis/Design
- Software Requirements Analysis
- Preliminary Design
- Detailed Design
- Coding & CSU Testing
- CSC Integration & Testing
- CSCI Testing
- System Integration & Testing

Operational Tests
- System Demo & System Qual Test
- System T&E, MEIT
- Engineering Test & Evaluation
- Systems Integration, Test and Performance Measurement

Development Phases
- System Requirements Analysis/Design
- Software Requirements Analysis
- Preliminary Design
- Detailed Design
- Coding & CSU Testing
- CSC Integration & Testing
- CSCI Testing
- System Integration & Testing

System Definition & Design
- Computer Program Definition & Design
- Computer Program Implementation
- Computer Program Testing
- System Integration Testing

Operational Reqts Definition
- Functional Breakdown (FB)
- Architectural Breakdown (AB)

System Reqts Definition

Element Reqts Definition

CP Performance Reqts Definition

Computer Program Design

Comp Program Detailed Design

Computer Program Design

Unit Test

Code

PRIME Responsibility

CSC Responsibility
Systems Integration

• Validate and maintain operability of system hardware and operating environments
• Verify Computer Program Interfaces

Hardware, OE, Computer Program Interface issues resolved during the computer program development phase
Systems Integration continued

- Integration issues tracking and reporting
  - Integration Issue resolution prior to computer program delivery

<table>
<thead>
<tr>
<th></th>
<th>BUILD ISSUES FOUND</th>
<th>ISSUES CLOSED DURING BUILD</th>
<th>DEFECTS WHICH BECAME TORs</th>
<th>ACTIVE ISSUES BEING ADDRESSED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BUILDS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build 4</td>
<td>18</td>
<td>15</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Build 5</td>
<td>53</td>
<td>37</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td>71</td>
<td>52</td>
<td>19</td>
<td>0</td>
</tr>
</tbody>
</table>
Systems Test

• Development and Regression Functional Testing
  – Documentation
    • Plans
    • Procedures
  – Execution
    • Utilizing two or more computer programs
      – Multiple hardware and operating environments

5% to 10% Improvement of Engineering Test & Evaluation
Systems Test continued

- Test defect reporting and tracking
  - Test defect resolution prior to computer program delivery
Systems Performance Measurement

• CPU and Memory Utilization
• Thread and Response Timing
• Measurement Tools
  – Legacy
    • ATES Data Recording
  – COTS
    • UNIX TOP and Kernel Trace
    • Concurrent Nightview
    • LynxOS Spyker
• Standard scenario
  – Function and information loading
  – Repeatable
Systems Performance Measurement continued

• Performance Measurement Abnormalities
  – CPU and Memory Utilization Increase
  – CPU Utilization Spikes
  – Memory Leaks
  – Slow Thread or Response
Systems Performance Measurement – CPU Utilization

User/System CPU Utilization

Time (Aug-19 Run)
Systems Performance Measurement – Memory Utilization

User Process Memory

Memory (Mem)

Time (Aug-29-Run)

- java
- *Sim* services
- system control
Systems Performance Measurement – Memory Leak

Free Memory

Time (Aug-29-Run)

Memory (Meg)

P1
Systems Performance Measurement continued

• Performance Measurement issues resolved before computer program delivery

10% to 20% CPU Utilization Improvement
Summary

• Incremental Systems Integration, Test and Performance Measurement
• Integration and Test issue and defect resolution during the computer program development phase
• Improved Computer Systems Stability and Performance

Cost Effective by identifying and resolving systems issues and defects during the computer program development phase
Tom Sobieralski  
Project Manager  
Computer Sciences Corporation  
304 West Route 38  
Moorestown, New Jersey  08057  
Voice: (856) 252-5052  
Email: tsobiera@csc.com