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

Tom Sobieralski
Computer Sciences Corporation
Agenda

• Background

• System Integration, Test and Performance Measurement

• Summary
Background
What is Computer Programs System 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 System Integration, Test and Performance Measurement

- Introduction of COTS hardware
- Conversion of legacy software to new languages
- System complexity with multiple hardware and operating environments
# Software Development Process

|----------------------------|--------------------------------------|----------------------------------|--------------------------|---------------------------|

## Development Phases
- Software Requirements Analysis/Design
- 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
- Element CP Integration & Test

## PRIME Responsibility
- CPAP
- MRP

## CSC Responsibility
- SDR
- PDR
- CDR
- DDR
- CR
- IPR

## Operational Reqts Definition
- FB
- AB

## System Reqts Definition

## Element Reqts Definition

## CP Performance Reqts Definition

## Computer Program Design

## Computer Program Detailed Design

## Code

## Unit Test
Multi-Processor Environments

• Legacy
  – AN/UYKs

• COTS Processors
  – Single Board Computers (SBCs)
  – Symetric 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
System Integration, Test and Performance Measurement
Process Improvement

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

CSC DMEI DE CMMI Level 5
Software Development Process
System Integration

• Verify Computer Program Interfaces
  – Interface Matrix

• Validate and maintain operability of system hardware and operating environments

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

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

<table>
<thead>
<tr>
<th></th>
<th>ISSUES FOUND</th>
<th>ISSUES CLOSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>31</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>5</td>
</tr>
<tr>
<td>Totals</td>
<td>58</td>
<td>11</td>
</tr>
</tbody>
</table>
System 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
System Test continued

• Test error reporting and tracking
  – Test error resolution prior to computer program delivery
System 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
System Performance Measurement continued

- Performance Measurement Abnormalities
  - CPU and Memory Utilization Increase
  - CPU Utilization Spikes
  - Memory Leaks
  - Slow Thread or Response
System Performance Measurement – CPU Utilization
System Performance Measurement – Memory Utilization

User Process Memory

Memory (Mem)

Time (Aug-29-Run)
System Performance Measurement – Memory Leak

Free Memory

Memory (Meg)

Time (Aug-29-Run)

System Performance Measurement continued

- Performance Measurement issues resolved before computer program delivery

10% to 20% CPU Utilization Improvement
Summary

• Incremental System Integration, Test and Performance Measurement
• Issue and Error resolution during the computer program development phase
• Improved Computer System Stability and Performance

Cost Effective by identifying and resolving issues during the computer program development phase
Tom Sobieralski
Project Manager
Computer Sciences Corporation
304 West Route 38
Moorestown, New Jersey 08057
Voice: (856) 252-2281
Email: tsobiera@csc.com