Abstract
The process described in the paper is being applied as part of the design verification of a replacement component designed for a nuclear weapon currently in the active stockpile. This process is an adaptation of the process successfully used in nuclear weapon development programs.

This paper describes a process to accomplish the system verification activity described in the Lockheed Martin Systems Engineering Process (LM-SEP). The verification process described in this paper addresses the system verification process for operational scenarios of the system during operational environments. The process is a complex combination of analysis, experimentation, computer modeling and simulations, and testing of physical hardware. The process described in this paper will concentrate on evaluating system response to radiation environments, verifying system performance during and after exposure to radiation environments, and assessing system survivability. The process begins early in the system development cycle and grows as knowledge and understanding increase during development and as the system design matures. The process culminates when the formal verification and validation results are presented to the Department of Defense’s (DoD’s) Design Review and Acceptance Group (DRAAG). However, post-deployment analysis and modeling may be required in conjunction with the surveillance activities conducted during the systems operational phase.

The system verification process described in this paper is an integral part of the LM-SEP. The verification process is part of the system verification activities in the LM-SEP Technical Process.
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
The verification process described in this paper can be thought of as a system. In fact, it closely follows the flow of activities described in the LM-SEP Technical Process. This verification process is being applied as part of the design verification of a replacement component for a nuclear weapon currently in the active stockpile. The process was used to successfully verify system responses and performance of nuclear weapon systems currently in the active nuclear stockpile. Although the process is currently being used to evaluate system design and response to hostile stockpile-to-target sequence (STS) environments, in the past the process was also used to evaluate the responses of nuclear weapon systems to normal and abnormal STS environments.

![LM-SEP Technical Process](image-url)

*Figure 1. The Technical Element of the Lockheed Martin Systems Engineering Process*

The system verification process is flexible. It can be adapted to the specific needs and constraints of the product system, the customer, and the available technology. The current implementation takes advantage of the advances in computer hardware and computational modeling and simulation being developed by the Department of Energy (DOE) as part of the Accelerated Strategic Computing Initiative (ASCI). This process is designed to compensate for the lack of experimental verification in the underground nuclear weapons effects tests at the DOE’s Nevada Test Site that was available to nuclear weapons development programs. This process is adaptable to a wide range of design verification activities.

During nuclear weapon development programs, Sandia National Laboratories conducted analytical, numerical, and physical evaluations of the systems responses to the possible electrical, mechanical, thermal, and radiation environments that could be encountered by the weapon system during its entire life cycle. These activities were reviewed on a regular basis by Sandia management and technical peers. In addition, Sandia frequently reported the results of those evaluations to subcommittees and working groups of the Project Officers Group (POG). Toward the end of the development program and before
quantity production, the results of the system evaluations (analyses and testing) were summarized and reported to the DoD’s DRAAG. The DRAAG approved the report and accepted the weapons.

All radiation environments that the weapon system might encounter during its life cycle are included in the evaluation process. These environments include those documented in the DoD’s STS document, space environments, and the additional radiation environment exposures to the weapon system during production and inspection of the warhead at DOE facilities.

**Systems and Subsystems**

The design verification process is applied separately to the system and to the subsystems that make up the system. This verification process is the last element of the LM-SEP technical process at each level of the systems engineering process (Figure 2). When appropriate, the verification process may be applied to subsubsystems and piece parts, but in a scaled-down manner. The systems engineering organization is responsible for the implementation of the verification process to the system. The objectives of the system-level verification process are to identify and address system-level issues and interface-related issues. Interface issues include physical or mechanical boundaries and electrical, data, and control interfaces. The interface issues include both the interfaces between the system and the external world, the interfaces among all of the subsystems, and the interfaces between each subsystem and the system.

![The Lockheed Martin Systems Engineering Process (LM-SEP)](image)

*Figure 2. The Lockheed Martin Systems Engineering Process (LM-SEP)*

Each subsystem supplier is responsible for implementation of the verification process for its subsystem. All of the individuals responsible for the verification processes must work effectively with each other to minimize overlaps and eliminate gaps.

The leader of the system-level process usually forms a panel of experts that reviews the system process and each of the subsystem processes. The reviews are known as peer reviews. The objectives of the
peer reviews are to (1) ensure that the processes are complementary and seamless, (2) validate each process, (3) verify the implementation of each process, (4) verify process documentation, and (5) document lessons learned for future programs.

The System Verification Process

Throughout the nuclear weapon development programs, the DOE design agencies (Los Alamos National Laboratory and Sandia National Laboratories) and DoD contractors work closely with the DoD customer’s team to refine and implement the verification process. Although the following list of steps for the verification process appears in numerical order, many are conducted concurrently. It is frequently necessary to iterate among several steps as the system design matures and as knowledge is gained. The verification process starts at the beginning of the system development process, which is usually thought of as the first four phases of the system life cycle (Figure 3). Early in the system development process, the verification process provides information for the trade-off studies. Later in the system development process, the system verification process provides information and feedback to the system designers and the design engineers. This paper describes the end product: the final system design verification process. This verification process is the end product of the knowledge gained during the development activities. However, it is not a separate activity; it reuses the computer models, material properties, and test data acquired during the development program.

Figure 3. The System Life Cycle

The objective of the design verification process is to verify the system design. In this paper, the process is described as if it were a stand-alone process. This appears to be a contradiction with the previous paragraph. However there is a good reason to present the process in this manner. The process evolves during the system development process to meet the needs of the system designers during the design phase. The process usually requires minor redirection and fine-tuning before design verification. A flow chart of the process is shown in Figures 4A, 4B, and 4C.
1. Define Operational Environments and Scenarios

2. Collect Relevant System Information

3. Decompose the System

4. Identify Critical Response and Damage Modes

5. Select Critical Operational Environments and Scenarios

6. Identify Computational Model Needs

7. Create Computational Models

8. Collect and Validate Material Properties

9. Determine Maximum Responses

10. Screen Response and Damage Modes

- Responses Exceed Known Thresholds? (Yes/No)
  - Yes: Notify system designers and return to Step 1 or 2.
  - No: Unvalidated and Unverified Computational Models

- Responses Very Low and Well Below Known Thresholds? (Yes/No)
  - Yes: Document rationale and eliminate from Process
  - No: To Step 11

To Steps 9, 13A, and 13C

Figure 4A. System Verification Flow Chart
11. Identify Damage Response Matrix Verification Activities

12. Design Experiments to Validate and Verify Computational Model

13A. Conduct Pretest Analyses

13B. Conduct Testing

13C. Conduct Posttest Analyses with computational models

14. Compare Model Results to Test Measurements

Acceptable correlation?

Yes

Validated & Verified Computational Models

To Step 15

Correct models and parameters

Figure 4B. System Verification Flow Chart- Continued
The design verification process should be validated and verified with the program manager and customer before implementation. We think of validation as “Are we doing the right things?” and verification as “Are we doing things right?” You’ll see this theme throughout the system verification process.
This verification process is strongly dependent upon past experience and existing knowledge. The ability of this verification process to detect and identify unknown or unexpected responses or failure mechanisms is limited.

The steps in this process are as follows:

1. **Define Operational Environments and Scenarios**

   The operational environments and operational scenarios are derived from the STS environments for the weapon system, the requirements described in the military characteristics document, and the anticipated operational deployment scenarios. Most of the information in the STS document is free-field data. Extensive analysis is required to identify the effects of the free-field STS environments on system and subsystem hardware. At the nuclear weapon systems level, a POG subcommittee usually coordinates this analysis. At the subsystem level, analysis is usually coordinated by the systems engineering organization. The analysis activity is typically a highly interactive and iterative process with extensive trade-off analyses and frequently results in a revised STS document. In addition, the products of this activity include documents that interpret and expand the information in the STS documents. This activity usually involves significant participation and cooperation of many DoD and DOE agencies and contractors.

2. **Collect Relevant System Information**

   Collect all of the system information that is relevant to the verification process. This includes design information, system functional operations and states, system performance requirements, operational deployment scenarios for the weapon system, customer-defined enemy defense system capabilities and engagement scenarios, material properties, and manufacturing process information.

3. **Decompose the System**

   Decompose the system to be verified into logical subsystems, assemblies, components, and piece parts convenient for analysis, computational modeling, and testing. The project manager for the system design verification project must coordinate the verification activities with the subsystem design verification projects. The project managers for the subsystem or component design verification projects must coordinate with the related subsystem design verification projects and the systems engineering organization. Subsystems and components are also subdivided into logical elements for the purpose of analysis, modeling, and testing.

4. **Identify Potential Response and Damage Modes**

   The objective of this step is to identify potential critical system and component response modes, damage modes, degradation mechanisms, and failure levels. This step is conducted by experts from the various participating organizations and agencies. The activities depend upon the experts' weapon system experience and knowledge supplemented by documentation from past weapon system programs. The failure levels identified for each damage and response mode, including references, should be documented.

   This step begins by reviewing the results of Step 3. Further subdivision or decomposition of the system, subsystem, or component into elementary parts may be necessary. Then the response and damage modes for each part are identified. As an example, a detonator consists of an external case, electrical conductors, electrical insulation, bridgewire, explosive pellet, welds, seals, and mounts. Any interactions or synergisms among parts are identified. In this example, an electrical charge created in
one part may flow through other parts to ground. The current flowing to ground could burn out another part in the path to ground.

Response and damage modes and degradation mechanisms are frequently categorized as electrical, mechanical, thermal, or radiation. Mechanical examples include stress, strain, plastic deformation, delamination, structural response, shock response, spall, cracking, work hardening. Electrical examples include degradation of transistor gain, threshold voltage shifts in field-effect transistors, burnout, latchup. Thermal examples include temperature changes, thermal expansion, phase changes, overaging. Radiation examples include neutron activation, damage to crystal lattices, discharge of capacitors. Degradation of transistor gain and threshold voltage shifts in field-effect transistors are examples of degradation mechanisms.

It is important to document the difference between damage thresholds and failure levels. A system may maintain full operational capability while it experiences many responses above the damage threshold. Failure, however, is usually used to refer to damage levels where full operation functionality is in doubt. As an example, electronic circuits are usually designed with sufficient margin to operate properly with radiation-induced damaged or degraded components. While there may be considerable damage to the individual components, the circuit is still functional and operational.

5. Select Critical Operational Environments and Scenarios

Identify the STS environments and operational scenarios that stimulate each of the potential system response modes that may interfere with the performance or reliability of the system. For hostile STS environments and enemy defense engagement scenarios, these responses are called vulnerability issues. The results of this step form the basis for determining the types and extent of computational models to be developed. Later, the results of this step will influence the experiments and testing. The results of this step must be verified and expanded once the computational models are available and validated. The main shortcoming of this process is that we cannot model what we don’t know.

Many STS hostile environments interact with hardware elements and cause vulnerability issues that must be addressed. On the other hand, many combinations of STS hostile environments and hardware elements do not cause vulnerability issues. Many hardware elements are subject to multiple vulnerability issues. The vulnerability issues are divided into mechanical issues and electrical issues for the analysis and testing activities.

6. Identify Computational Model Needs

The objective of this step is to identify computational models that must be created for the system design verification process to succeed. The requirements for the computational models are also developed during this step. The computational models are used to simulate the systems response to the environments specified in the STS normal, STS abnormal, and STS hostile environments. A partial list of the computational models required for simulating system performance in hostile STS environments include the following:

- Full-body structural response (air blast, x-ray-induced impulse, and x-ray-induced thermostructural response [TSR])
- Thermal (combined hostile STS environments, ascent and reentry heating, thermal battery heat, heat from the decay of radioactive materials in the nuclear package, etc.)
- Electric/electronic circuit analysis (radiation-induced response)
- Radiation transport of STS environment into system
- X-ray-induced thermomechanical shock (TMS) response
- Neutron effects
- Electromagnetic pulse (EMP), systems-generated EMP (SGEMP), cable systems-generated EMP, internal EMP (IEMP), transient radiation effects on electronics (TREE)
- Hostile electromagnetic radiation (EMR)
- Single-event upset
- Latchup
- Reentry aerodynamics
- Air blast flow fields
- Combined environments modeling
- Multiple hostile encounter modeling

7. Create Computational Models

Computational models (numerical simulations) of the system are created for simulating hardware response modes and damage modes that occur when the system is subjected to operational environments. Often the computational models of the system are constructed by combining computational models of the subsystems. The subsystem models are usually validated separately in a series of numerical simulations and physical environmental tests. When appropriate, model development and geometry are coordinated with other members of the customer's systems development (design) team. Configuration management and change control are applied to elements of the system computational models early in the development process, otherwise, there will be endless arguments at design reviews about whose model is “right.”

A. Simulating System Response to Environments. Computational models are created to numerically simulate the response of the system to environments applied externally to the system. They are also used to numerically simulate the response of the system to internally generated environments. Examples of externally applied environments include weather, lightning, thermal cycles, transportation environments, deployment environments, and defensive countermeasures. An example of an internally applied environment is the heat from a thermal battery.

B. Creating Computational Models. Computational models are created by combining a computer software tool with physical information describing the system. Examples of computer software tools are finite-element analysis, thermal analysis, and radiation transport codes. The physical descriptions of the system are usually approximations of the physical dimensions, shape, material composition, material properties, joints, and boundary conditions. An important aspect of the validation of the computational models is to assess the quality of these approximations.

C. Validating and Verifying Software Tools. Validation and verification of the software tools should occur outside the scope of the system verification process described in this paper. When unvalidated and unverified software tools are needed, their validation and verification should be included in the system verification process. A validated software tool contains the correct or appropriate physics. In a verified software tool, the physics is correctly implemented, there are no programming or coding errors, and the numerical analysis techniques are appropriate for this application (converge to the correct solution).
D. Validating and Verifying Computational Models. In a validated computational model, the definition of the system's geometric features, material properties, and boundary conditions are implemented with sufficient detail and accuracy to allow meaningful numerical simulation of the system performance. A validated computational model correctly simulates the proper system response modes in known test cases with known geometry and in known environments. All the material properties and parameters used in the computational model should be properly validated throughout the entire range of the simulation (i.e., temperature and pressure ranges). Material properties that are not validated should be validated as part of the system verification process. In a verified computational model, the system description information and boundary conditions are correctly incorporated into the computer code. Later in this process, considerable effort and resources are devoted to validating and verifying the computational models.

In the future, improved computational modeling capability and computer platforms will permit better modeling and simulation of nonlinearities and boundary conditions as well as incorporate more complete representations of the physics. An important goal of the verification process is for the modeling and simulation activities to keep up with the design activities and changes. In the past, this has been difficult because of the large amount of effort required to remesh the computer models for sometimes very slight changes in the design details.

8. Collect Validated Material Properties

A validated set of material properties, input parameters, damage thresholds, failure levels, and component information is obtained by literature searches and searches of databases from previous design verifications projects. All material properties, damage thresholds, and failure levels should be validated throughout the entire range of the computational model simulations. Missing information can be created and validated by analysis and experiments. The organization responsible for implementing the design verification process may need to sponsor experimental approaches to obtain the necessary information. The plan for obtaining much of the missing information is described in the verification compliance matrix (VCM) discussed later in this paper.

When appropriate, the model parameter definitions and values should be coordinated with appropriate members of the customer’s system development team. Early in the system development process, apply configuration management and change control to the model input parameters. The customer usually appoints a keeper of the material properties library; the library is shared by all of the participants and design agencies.

9. Determine Maximum Responses

The objective of this activity is to determine maximum or worst-case magnitudes for each response mode, damage mode, and the associated operational STS environment that produces it. There are three primary reasons for this activity. First, if any responses are near or exceed known failure levels, elements of the system may need to be redesigned. Second, the computational models should be validated and verified at the extreme ranges of the responses, and third, sensitivity studies should be conducted near the maximum responses. Improved modeling capability and computational platforms will greatly aid this activity. This activity will be conducted with the best available computational models. However, at this point in the process, most computational models will not be validated or verified. The results of these activities should be verified or revised when the validated and verified computational models are available.
Initial sensitivity studies are conducted at the locations where the computational models predict maximum responses. Later in the process, when validated and verified computational models are available, more thorough sensitivity studies will be conducted. Typical sensitivity analyses would consider variations in material properties, material composition, material dimensions, and external environments.

The first part of this activity considers a single STS environment or a single-enemy defensive measure. Concurrent environments or multiple enemy countermeasures are possible. It may be necessary to address the issues associated with combinations of STS environments or multiple enemy hostile encounters.

10. Screen Response Modes, Damage Modes, and the Associated Radiation or Hostile STS Environments

The system responses are divided into three categories. The first are those responses where known damage or failure thresholds are exceeded to the point where system performance or survival are in serious doubt. The second are those responses that are so far below known damage thresholds that further analysis or testing is not justified. And the third are all the other cases. The items in the first category should be discussed with the system designers. Design alternatives may be suggested when appropriate. These issues can be dealt with by either design modifications or restrictions on the STS environments. If a design modification or an STS change is necessary, notify the system and design groups immediately and return to Step 1 or 2 to include the design modifications in subsequent analyses.

The items in the second category should be removed from further consideration. The screening decisions will be verified later in the verification process, when the computational models are verified and validated. These items are the "uninteresting" cases in which (1) the responses are well below known damage thresholds and in which there are no known synergistic combinations and (2) damage thresholds are exceeded, but the damage has negligible impact on reliability or system performance. Justification for discarding these cases should be well documented. The discarded responses will be reevaluated (validated) later (Step 15) in the verification process. The screening criteria may be suitable for prioritizing the remaining issues. The manager of the verification process may choose to prioritize the remaining issues during this step.

The screening activity draws heavily on past experience, the confidence associated with those past experiences, and data from past systems verification projects. If prioritization of the issues is needed, one of the criteria should be the confidence the analysis team has with the quality and applicability of past experiences and data from previous projects.

The results of the screening activity are documented on a damage-response mode matrix (DRM) as illustrated in Table 1. Each intersection of response and damage mode is a potential vulnerability issue. The credible vulnerability issues (damage modes) are identified in Step 4. Every credible damage mode is identified with a unique DRM identification number in the first column for traceability purposes. Calculated response levels are compared with known or estimated failure levels to establish a safety margin. The safety margin is used to rank order or screen the damage modes and is listed in the upper part of each credible damage/response intersection. Color coding is useful for ranking the credible damage/response intersections into the three system response categories. Red is used for safety margins less than one. This corresponds to the first category where system survival or performance are in serious doubt. Green is used for those items in the second category with safety margins greater than five where no further consideration or analysis is justified. Yellow is used for the third category with safety factors greater than one, but less than or equal to five. The lower portion of the DRM intersection cell is used to
uniquely identify a set of verification activities to further refine the response, damage, and failure levels. Interface issues among the subsystem or among the subdivided must be addressed in the DRM. In addition, the DRM must include system issues with the external world and system issues with the subsystems.

### Table 1. Example of a Damage-Response Mode Matrix

<table>
<thead>
<tr>
<th>DRM I.D.</th>
<th>Path(s)*</th>
<th>Damage Mode</th>
<th>Response Modes (Safety Factor/Verification I.D.)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-1</td>
<td>1, 5, 7</td>
<td>Internal component damage due to shock transmitted from cover</td>
<td>2</td>
<td>Need shock properties of tin</td>
</tr>
<tr>
<td>X-2</td>
<td>3, 3a, 6</td>
<td>Discontinuity of ground return path due to spall/delamination of cover</td>
<td>?</td>
<td>Need delamination threshold of SiC and thermal analysis</td>
</tr>
<tr>
<td>X-3</td>
<td>1, 4</td>
<td>Failure of internal components due to late-time responses</td>
<td>0.9</td>
<td>Need TSR calculation</td>
</tr>
<tr>
<td>X-4</td>
<td>all</td>
<td>Failure of connector</td>
<td>4</td>
<td>Peak $\Delta T$ at interface is 200 °C</td>
</tr>
</tbody>
</table>

* The worst case path is underlined.

**Legend:**
- Safety Margin: Safety Margin < 1 | Safety Margin between 1 - 5
- Verification Task Status: Task Not Started | Task Under Way | Task Complete

Blank cell means no issue was identified.

### 11. Identify Damage Response Matrix Verification Activities

Identify tests, analyses, or other verification activities that will be used to verify compliance with the requirement for each credible damage mode. Assign a unique code to each verification activity as shown in the lower part of each cell in Table 1. Create a verification compliance matrix (VCM) as illustrated in Table 2. A description of each verification activity is then created in Table 3. These VCM activities form the basis for identifying the computational model validation issues and the corresponding tests.
Table 2. Example of a Verification Compliance Matrix

<table>
<thead>
<tr>
<th>Verif. I.D.</th>
<th>Requirement</th>
<th>Verification Activities</th>
<th>Design Limit</th>
<th>Response</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XV-1</td>
<td>Transmitted shock &lt; design margin of internal components</td>
<td>Tests: XA-4</td>
<td>4 kbar</td>
<td>2 kbar</td>
<td></td>
</tr>
<tr>
<td>XV-2</td>
<td>Maintain electrical continuity of cover</td>
<td>Analyses: XA-2</td>
<td>?</td>
<td>3.3 kbar</td>
<td>Test of SiC spall strength required</td>
</tr>
<tr>
<td>XV-3</td>
<td>Applied loads &lt; design margin of internal components</td>
<td>Exams: XA-3</td>
<td>0.9</td>
<td>1</td>
<td>Oscillator may fail in late-time</td>
</tr>
<tr>
<td>XV-4</td>
<td>Peak tensile stresses &lt; spall strength of connector</td>
<td>Test: XA-5</td>
<td>8 kbar</td>
<td>1 kbar</td>
<td>8 kbar spall strength based on engineering judgement</td>
</tr>
<tr>
<td>XV-5</td>
<td>Thermal conduction assessment (2-D, 3-D)</td>
<td></td>
<td></td>
<td></td>
<td>Need thermal analysis</td>
</tr>
</tbody>
</table>

Legend:

- Task Not Started
- Task Under Way
- Data Obtained (Test)

Blank cell means no verification activity.
### Table 3. Example Verification Activities

**XV-1: Transmitted Shock < Design Margin of Internal Components, Path 5**

<table>
<thead>
<tr>
<th>Task I.D.</th>
<th>Task Title</th>
<th>Test Series</th>
<th>Assigned To</th>
<th>Priority</th>
<th>Status (% Complete)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XA-1</td>
<td>Tensile stress at internal component in</td>
<td>N/A</td>
<td>DLR/PCR</td>
<td>1</td>
<td>100</td>
<td>WONDY</td>
</tr>
<tr>
<td>XT-1</td>
<td>Ultimate strength of materials tin and SiC</td>
<td>L98-1</td>
<td>WHB</td>
<td>1</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>XT-2</td>
<td>Spall strength of tin</td>
<td>H98-2</td>
<td>EDS</td>
<td>1</td>
<td>100</td>
<td>Electron beam exposure</td>
</tr>
<tr>
<td>XT-3</td>
<td>Spall strength of SiC</td>
<td>H98-2</td>
<td>EDS</td>
<td>1</td>
<td>100</td>
<td>Electron beam exposure</td>
</tr>
</tbody>
</table>

**XV-2: Maintain Electrical Continuity of Cover, Path 6**

<table>
<thead>
<tr>
<th>Task I.D.</th>
<th>Task Title</th>
<th>Test Series</th>
<th>Assigned To</th>
<th>Priority</th>
<th>Status (% Complete)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XA-2</td>
<td>Tensile stress in cover</td>
<td>N/A</td>
<td>DLR/PCR</td>
<td>2</td>
<td>50</td>
<td>WONDY</td>
</tr>
<tr>
<td>XT-4</td>
<td>Delamination threshold of SiC/tin interface</td>
<td>H98-3</td>
<td>EDS</td>
<td>2</td>
<td>75</td>
<td>Electron beam exposure</td>
</tr>
</tbody>
</table>

**XV-3: Applied Loads < Design Margin of Internal Components, Path 16**

<table>
<thead>
<tr>
<th>Task I.D.</th>
<th>Task Title</th>
<th>Test Series</th>
<th>Assigned To</th>
<th>Priority</th>
<th>Status (% Complete)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XA-3</td>
<td>3-D TSR in internal component</td>
<td>N/A</td>
<td>DLR/PCR</td>
<td>1</td>
<td>24</td>
<td>PRONTO3D TSR calculation</td>
</tr>
</tbody>
</table>

**XV-4: Peak Tensile Stresses < Spall Strength of Connector, All Paths**

<table>
<thead>
<tr>
<th>Task I.D.</th>
<th>Task Title</th>
<th>Test Series</th>
<th>Assigned To</th>
<th>Priority</th>
<th>Status (% Complete)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XA-4</td>
<td>3-D stress in connector</td>
<td>N/A</td>
<td>DLR/PCR</td>
<td>5</td>
<td>100</td>
<td>PRONTO3D TMS calculation</td>
</tr>
<tr>
<td>XT-5</td>
<td>Spall strength of connector materials</td>
<td>S98-1</td>
<td>WHB</td>
<td>5</td>
<td>100</td>
<td>Plasma radiation source exposure</td>
</tr>
</tbody>
</table>

**XV-5: Thermal Conduction Assessment, Path 6**

<table>
<thead>
<tr>
<th>Task I.D.</th>
<th>Task Title</th>
<th>Test Series</th>
<th>Assigned To</th>
<th>Priority</th>
<th>Status (% Complete)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XA-5</td>
<td>Thermal conduction assessment in 2-D, 3-D</td>
<td>N/A</td>
<td>DLR/PCR</td>
<td>2</td>
<td>0</td>
<td>COYOTE</td>
</tr>
</tbody>
</table>

12. **Design Experiments to Validate and Verify Computational Models**

The objective of this activity is to design experiments and physical system tests that will validate and verify the computational models used in the system verification process. Good computational models are a requirement for designing good experiments and tests. A subset of the response modes and damage modes derived by the unvalidated computational models and identified in Step 9 and passing through the screen in Step 10 must be selected for comparison between the models and the tests.

**A. Responses Excited in Tests.** The response modes and damage modes selected above will be measured in the experiments and tests. The objective of these experiments and tests is to collect data for comparison and correlation with critical elements of the computational models. The goals of these comparisons include validating the computational models, improving details of the parameters within the computational models, and improving the detail of the submodels where necessary and feasible (e.g., better modeling of interfaces and nonlinear joints).
B. **Identify Test Environments.** The objective of this activity is to identify acceptable test environments that will stimulate* the correct critical system response modes and damage modes in the test hardware. The unvalidated computational models are used to determine the test environments and forcing functions that stimulate or drive the desired response modes to the magnitudes desired in the tests. Also, identify the measurements of the test environments needed to verify that satisfactory test environments and test conditions were achieved during the test. These levels should be near the maximum magnitudes expected in the system hardware during operational scenarios in operational environments.

C. **Select Sensor Locations.** The existing, unvalidated computational models play a key role in defining the tests that will validate and verify the computational models. The validation and verification activities must be carefully planned and implemented to avoid circular arguments and faulty logic. Simulations with the unvalidated computer models identify the portions of the system hardware where the maximum levels of the response modes and damage modes identified in paragraph B occur. This information is used to select locations for the instrumentation sensors. Often, physical constraints prevent locating the sensors at the precise location where the maximum response occurs. The sensors should be placed as near as practical to the maximum response point. The pretest analysis, used to set up the data acquisition system, should include the exact sensor location.

D. **Select Sensor Type and Sensitivity Range.** Computational predictions of the maximum response levels and locations of the maximum responses derived in the previous paragraph guide the selection of sensor types, sensor sensitivity ranges, instrumentation system frequency ranges, data filtering, data amplification, and data recording requirements. These predictions will also be used to determine the initial instrumentation system setup parameters. The sensors should be calibrated before installation into test hardware. Sensor response and performance may be different in specific test environments than described in the manufacturer’s literature. These differences must be factored into the instrumentation setup and calibration as well as the posttest processing and analysis of the data. Calibration of the complete data acquisition system is normally required before the test. The characteristics of the data acquisition system that might limit the accuracy or resolution of the data, such as bandwidth limitations and unexpected filtering of the data must be detected prior to conducting the tests. Finally, requirements for the posttest processing and analysis of test data must be defined.

E. **Define Test Hardware and Modification.** The quality and usefulness of the data collected in a test depends upon the quality of the system hardware used in the test. The hardware selected for the test must be capable of faithfully reproducing the desired response modes and damage modes. This is an important point. Hardware is expensive and scarce during development programs. Often, hardware rejected from other activities can be suitable for tests. Early in the definition of the test program, hardware needs, including maturity of hardware design and modifications, must be carefully and thoughtfully defined. It is frequently necessary to modify the test hardware to accommodate instrumentation sensors and cables or to simulate hardware conditions during an actual STS engagement (i.e., thin heat shield and nose to account for reentry ablation).

---

*This approach *stimulates* the response modes expected in a hostile STS engagement rather than simulating the STS environments. In other words, we are rarely able to directly simulate the STS environments with the test environment. It is also assumed that the computer software (such as finite element, radiation transport, or shock physics codes) used to construct the computer model of the system has been adequately verified and validated.
F. **Define Posttest Procedures.** The posttest procedures and activities should be thoroughly thought out and identified during the test planning activities. The posttest activities involving the test hardware, such as measurements, inspections, photographs, materials analysis, material tests, instrumentation tests, recovery procedures, and documentation requirements should be defined. The posttest data processing and analysis procedures should be defined before the test.

G. **Verify Test Environments.** The system responses are a direct function of the test environments and loads placed upon the hardware by the test environments. Accurate comparisons of the results from simulations using the computational models with the test data require the use of measured test conditions and environments as inputs to the computational models. Measurements must be made to verify that satisfactory test environments and test conditions were achieved during the test. These measurements must provide sufficient input data to the computational models to enable simulation of the response and damage modes excited in the system hardware during the tests. During the planning activities, we must define measurements and instrumentation to validate and verify environments applied to the system hardware during the test.

13. **Conduct Pretest Analyses, Testing, and Posttest Analyses**

The objective of this step is to collect data for validating and verifying the computer models. This step includes collecting system hardware and instrumentation for the test, modifying the system hardware as needed, installing the instrumentation, conducting the test, verifying the test conditions and environments, and verifying the data collected during the test. The test environments are measured during the test and the results are compared to the desired test environments. After the test, the data are collected and inspected for consistency among sensors and consistency with the appropriate physics. The data are also inspected for signs of sensor or recording system failures. After the test, a thorough analysis and inspection of the system test hardware and instrumentation system are made. The system test hardware should be inspected, photographed, and videotaped *in situ* before moving or recovery of test hardware. The pretest activities, the test, and the posttest recovery activities are documented in the test plan.

Acceptance inspections and tests are needed before accepting the system hardware for the test. The system test hardware must be suitable for the test if it is to be accepted. After the system hardware is accepted, it can be modified as needed for the test, and the instrumentation sensors can be installed. Assembly of the system test hardware follows a well-thought-out set of written assembly (and disassembly) procedures. If possible, videotape the assembly and installation of the instrumentation. Preshipment inspections and tests should be conducted before shipping the instrumented system test hardware to the test facility. Immediately before executing the test, inspect (verify) and photograph the test setup and the system test hardware in the final test configuration. These activities, inspections, calibrations, measurements, and documentations should be identified in the system verification plan and test plan.

The unvalidated computational models are used to predict maximum responses at the actual sensor locations. This prediction is done before the actual test. The computational models should use the exact specifications of the test hardware, including any modifications made to accommodate the instrumentation. The results of these simulations are used for final setup and fine tuning of the data acquisition system. The details in these computational models are usually slightly different than the actual system hardware to account for the actual test setup. The same computational models will be used to predict the posttest response using the measured test environments.
14. **Compare Computational Models to Tests Results**

The unvalidated computational models are used to calculate the posttest response modes, and damage modes expected in the test hardware during the test. The computational models used for the posttest simulations should be identical to the models used for pretest analysis, with the exception that the posttest simulations will use the measured test environments. The computational simulations of the test hardware responses incorporate the exact configuration of the test hardware, including modifications made to accommodate instrumentation, mounting the hardware in the test facility, and the measured test environments and loads applied to the test hardware.

If measurable damage or failures have occurred in the test hardware as a result of the test, additional numerical simulations may be required to find the set of input parameters that bracket the damage or failures.

Compare the results of the posttest simulations with the test data at all the test hardware sensor locations. It is not unusual for the agreement between the computational simulations and the test data to vary substantially at locations across the system model and for there to be better agreement with some response modes than with other modes.

It is now time for the system analysts to understand and account for the differences. The differences may be traceable to the model or they may be traceable to the test data. What lies ahead is a challenging and time-consuming exercise. Typical sources for these differences within the model include misunderstandings of the modifications made to the test hardware, insufficient approximations of the system hardware made during model creation, inadequate specification of boundary conditions, insufficient mesh structure, inadequate specification of material parameters, or inaccurate values for material parameters. Typical sources for these differences within the test data include unanticipated electrical filtering of the test data, improper installation of sensors, damage to sensors, improper calibration of the instrumentation channel, or assigning the wrong data record to a sensor.

The analysts and model builders will continue working with the computational models until the agreement between the test data and the posttest computational simulations is deemed adequate for verification of the system. At this point, the computational model is considered to be validated and verified. This procedure must be followed for most of the models listed in Step 6.

The difference may be of such a nature that it may be necessary to return to Step 12 and conduct additional tests and computational simulations. This may be necessary for the entire system or may be necessary only for selected subsystems. Even worse, it may be necessary to cycle back and repeat Steps 7 through 13. Usually when it's necessary to cycle back to Step 7, it's only for one or a few subsystems. In past system development programs, system tests were conducted on development hardware during the full-scale engineering development phase. The major difficulties with the computational models and the testing procedures were identified and corrected in the development hardware. During the verification phase, system tests with production-quality hardware were usually in good agreement with the test data (that is, as good as possible with the computer technology of the day), so there was usually little modification of the computational models in the verification phase.

15. **Verify Screening in Step 10**

Use the validated and verified computational models to validate the decision to delete the scenarios and STS environments discarded in Step 10. If simulations with the validated computational models cannot justify the omission of some of the cases in Step 10, add them back into the analysis and modeling.
16. Use Validated Computational Models to Simulate System Response in all STS Environments

The validated computational models are used to simulate (predict) the system response and failure modes in the hostile environments and STS scenarios identified in Step 1. It is not practical to analyze all of the possible STS scenarios; thus, a subset of the STS scenarios and environments must be chosen. The subset usually includes environments and scenarios that encompass or envelop the possible scenarios and environments in the STS. This activity also verifies the results of Steps 4 and 5. This step usually includes the specific analysis and simulations with the validated computational models called for in the VCM.

The activities in this step are divided into three parts. First, the response of the system to single STS environments is simulated. This is done by considering each environment separately. Second, the response of the system is simulated and evaluated to combinations of the STS environments. Many combinations of STS environments cannot occur in operational scenarios. These analyses and simulations are limited to legitimate combinations of STS environments. The third part of this step is more difficult. The systems response in an STS environment early in the operational scenario may cause degradation or damage to the system hardware. The extent of degradation or damage may have no adverse effect on system operation or performance. The objective of this part of the step is to make sure that this damage or degradation does not couple with or become worse in a subsequent STS environment.

17. Conduct Sensitivity Analyses

Conduct sensitivity analyses relative at the points of maximum responses or at physical locations and environments near known damage thresholds or failure levels.

18. Compare System Responses to “Acceptable Limits”

Compare the system response levels calculated in Steps 16 and 17 to the known “damage levels and acceptable limits” documented in Step 8 and determined by experiments identified in the VCM. These levels are either documented in Step 8 or are derived in the experiments sponsored by this system verification process. The experiments are summarized in the VCM.

If system response levels calculated in Steps 16 and 17 are greater than known “acceptable limits,” alert the system design group and appropriate POG subcommittees and working groups as soon as possible and recheck analysis, input parameters, computer models, interpretation, and scaling of the test data. If any responses exceed acceptable limits, redesign of the system hardware or revision of the STS requirements is likely required. In this event, parts of this process will need to be repeated.

19. Verify the Process and Results with the DoD Customer, the DOE, and Peer Reviewers

The system evaluation process should be validated by the DoD system customer, DOE representatives, POGs, and peer reviewers before beginning Step 1. This process was used successfully for the verification of the existing nuclear weapon system in hostile environments, so it has been validated to some extent. One would expect that with today’s computers, modeling capability, and lack of test facilities, the process would rely more on modeling and less on testing in the future. Lessons learned and improvements made during implementation of the verification process should be documented.
20. Document the Results

The activities of the verification processes should be thoroughly documented. Decisions, screenings, material properties, hardware vintages, model details, and results should be included in the documentation. The documentation usually contains a series of reports describing the modeling and testing. The results of the verification process are summarized in the final report.

Documentation of this Verification Process

During weapon development programs, the activities and the results of this system verification process were documented. However, to the authors’ knowledge, the overall verification process was never documented. It is recommended that in future weapon programs the system verification process be thoroughly thought out and defined, validated with the customer, and documented before starting major analysis, modeling, and testing activities.

Author Biographies

Frank Dean is currently a manager at Sandia National Laboratories. He has been at Sandia since 1965. He has worked in the New Mexico Weapons Systems Engineering Center since 1982. He is a member of the EPI Systems Engineering Subcouncil, INCOSE, and PMI. He has a B.S. and M.S. in Mathematics from Illinois Institute of Technology.

William Barrett is currently a principal member of the technical staff at Sandia National Laboratories. He has worked at Sandia since 1980. He has been engaged in nuclear weapons effects studies including nuclear underground testing, above-ground simulation testing, and nuclear effects hardness assessments. Prior to working at Sandia, he was a Nuclear Research Officer in the U.S. Air Force where he had similar responsibilities from 1971 until 1980. He has a B.S. in Mechanical Engineering from the University of Wyoming and an M.S. in Nuclear Engineering from the U.S. Air Force Institute of Technology.

References

2. The Department of Energy's Development and Production Manual, AL 56XB.

Acknowledgements

The system verification process has evolved over several decades. Significant contributors to the process include Lockheed Martin Missiles and Space, Sunnyvale, California; ITT Systems and Sciences, Colorado Springs, Colorado; Ktech, Albuquerque, New Mexico; Los Alamos National Laboratory; and several DoD organizations.

The authors wish to thank Systems Engineering Subcouncil members Sandy Fridental and Tom Duffy for helping us understand the Lockheed Martin Systems Engineering Process and how it fits into our work.