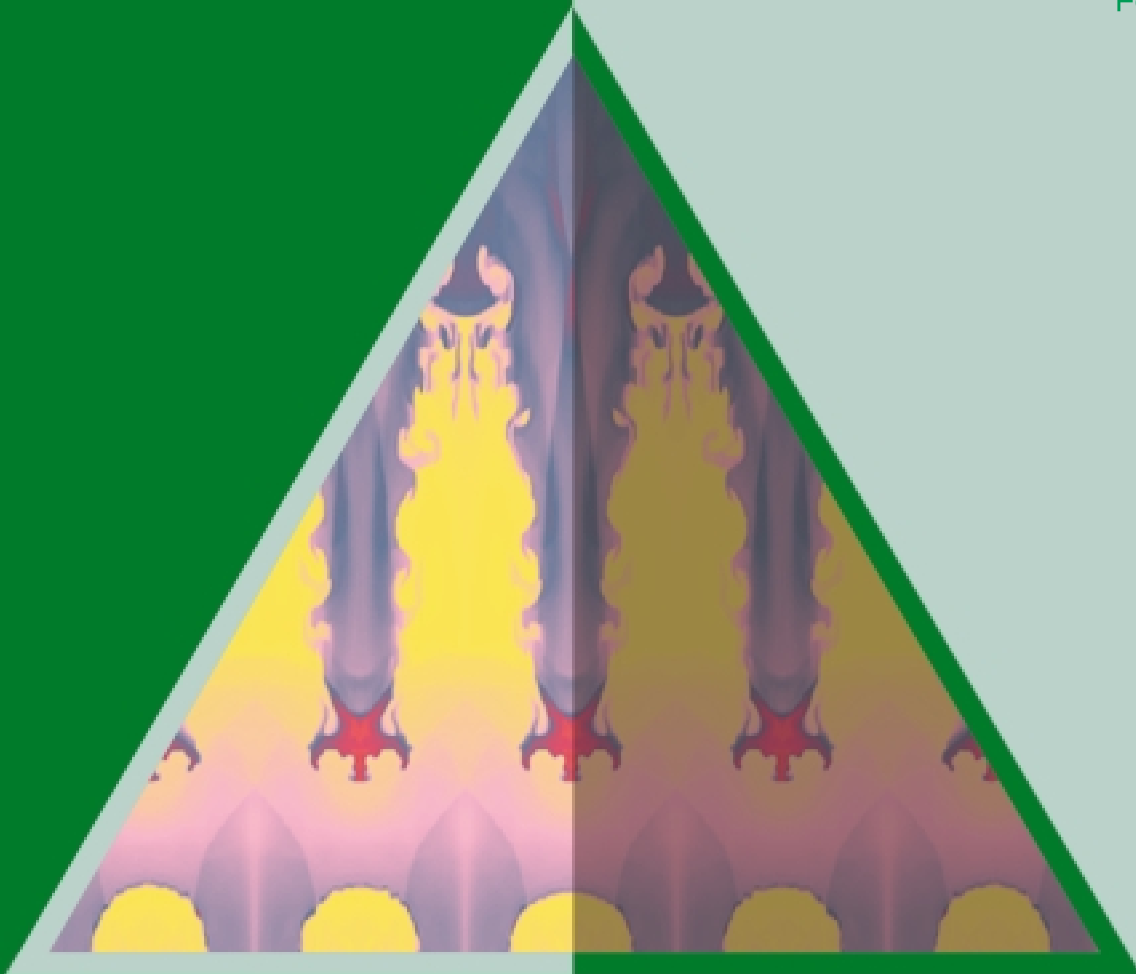
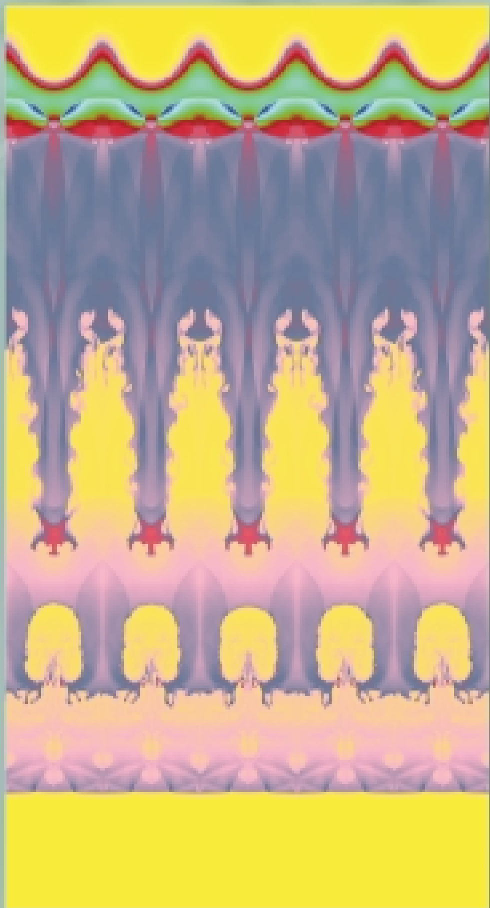
DOE/DP/ASC –
SQE-2000-FDRFT-VERS2
February 2001



ASCI

Software Quality Engineering:

GOALS, PRINCIPLES, AND GUIDELINES



ON THE COVER:

ASCI at Livermore National Laboratory:
A Hydrodynamic Instability Calculation

GRAPHIC DESIGN AND PRODUCTION

Michael Vittitow - Creative Arts Department-12620, Sandia National Laboratories, Albuquerque, NM.

ASCI
Software Quality Engineering:

GOALS,
PRINCIPLES,
AND
GUIDELINES

AUTHORS

SANDIA NATIONAL LABORATORIES

Ann Hodges and Gary Froehlich
Information Systems Engineering Center

David Peercy
Surety Assessment Center

Martin Pilch
Engineering Sciences Center

Juan Meza
ASCI Office – Special Assignment

LOS ALAMOS NATIONAL LABORATORY

Maysa-Maria Peterson and Jeanette LaGrange
Scientific Software Engineering

Lawrence Cox
Applied Physics Division

Ken Koch
Nuclear Weapons Program Office

LAWRENCE LIVERMORE NATIONAL LABORATORY

Nancy Storch
Scientific Computing Applications Division

Cynthia Nitta and Evi Dube
Defense and Nuclear Technology Directorate

ACKNOWLEDGEMENTS

The authors would like to thank the following individuals for their thoughtful comments. There are many other individuals who have provided comments as part of group reviews and reviews of which we undoubtedly are not aware. Every comment was carefully considered and was either incorporated, or influenced the final version of this document.

At Sandia National Laboratories:
Ed Boucheron, Steve Kempka, Mark Kiefer, Dwayne Knirk, Jaime Moya, Bill Oberkampf, James Peery, Bob Thomas, Tim Trucano, and John Zepper.

At Los Alamos National Laboratory:
Gregg McKinney, Bill Rider, and Tom Seed

At Lawrence Livermore National Laboratory:
Linnea Cook, Richard Klein, Roger Logan, and David Sam.

CONTENTS

ABSTRACT

This document describes the goals, principles, and guidelines for software quality engineering (SQE) activities of the Accelerated Strategic Computing Initiative (ASCI) program. It is expected that site-specific organizations and projects supporting the ASCI program would satisfy these SQE guidelines through their own software practices, established using a prioritized and graded approach, and leveraging common best practices and established practices. The benefit to the ASCI program is dependent on how well these deployed practices satisfy certain principles considered important to fulfilling the ultimate goals of establishing confidence in the ASCI codes and enhancing the credibility of ASCI modeling and simulation results.

This document focuses primarily on SQE – a part of the overall ASCI Verification and Validation (V&V) Program. Many of SQE's principles and goals are the same as those of V&V; however, the emphasis is solely on software-related issues. This document, therefore, provides guidelines strictly for software verification, software engineering, and project management activities, as they apply to SQE in general. How organizations and projects integrate these activities by using effective practices and work products depends on the use of their own methods and techniques.

SQE practices should be improved according to these principles and guidelines, and in a prioritized manner as determined by the ASCI program and development teams. Software process improvement plans should consider current best practices of the ASCI development teams, and the guidelines in this document provide recommendations for further process improvements.

EXECUTIVE SUMMARY

The focus of the Stockpile Stewardship Program (SSP) is to maintain high confidence in the safety and reliability of the nuclear weapons stockpile in the absence of nuclear testing. In 1995, the Accelerated Strategic Computing Initiative (ASCI) was established as a critical element to help shift from test-based confidence to science- and simulation-based confidence. A key component of ASCI is the Verification and Validation (V&V) program which seeks to provide high confidence in the computational accuracy of ASCI simulations by systematically measuring, documenting, and demonstrating the predictive capability of the codes and their underlying models.

An important component for developing increased confidence in ASCI simulations is a clear and consistent set of guidelines that applies to all ASCI software projects. These guidelines are envisioned as part of an overall verification and validation process for each software project. Within the ASCI program[1], verification is defined as the process of determining that a computational software implementation correctly represents a model of a physical process. Similarly, validation is defined as the process of determining whether a computer model correctly represents the real world from the perspective of the intended uses of the model or simulation. In a broader context, however, the modeling and simulation of an application can be viewed as a hierarchy of representations, starting with a conceptual physics model, to mathematical algorithms, to numerical algorithms, and finally resulting in a software implementation. In this broader sense, verification consists of those activities that ensure the requirements of one representation are implemented accurately by a succeeding representation. As related to software, however, verification consists of those activities that determine the extent to which the development process used sound and established software engineering techniques. Software validation is focused on delivering an appropriate software product to the user, accurately represented by the user's requirements and by functionality in the code.

[1]Verification and validation definitions were extracted from: DOE/DP-99-000010592, ASCI Program Plan, January 2000

Within the ASCI program, these latter software verification and validation definitions have been associated with software quality engineering (SQE).

This document is a composite of agreements among the three DOE defense laboratories of goals, principles, and guidelines for ASCI's SQE activities. The purpose is to document and enhance current SQE practices for all ASCI software projects. The selection of specific practices will depend on many factors. It is expected that the practices will be tailored to each individual organization, and perhaps even to each major code project. For example, in a production environment it is expected that more rigorous practices would apply than for an exploratory or research environment.

The goal of the ASCI V&V program is to provide adequate confidence that the stockpile-certification calculations using the advanced three-dimensional applications developed by ASCI have a well-understood range of applicability and that uncertainties in the results have been analyzed and quantified. To fulfill this goal, the V&V program verifies codes and validates activities intended to (a) produce easily maintained and error-free software packages, and (b) to ensure that the codes give an adequate simulation of physical reality. Supporting this goal is the following set of principles applicable to all ASCI software projects.

- **Fidelity** – accuracy and correctness of simulation results.
- **Functionality** – the degree to which a simulation fulfills user requirements.
- **Repeatability** – the capability to reconstruct simulation results.
- **Traceability** – the capability to view the progress of a project, and the links and relationships among associated work products.
- **Manageability** – the capability for management to determine a project's status, progress, and capabilities.
- **Supportability** – the adequacy of resources and procedures to facilitate changes to, and update releases of, work products.

Guidelines to support the principles were derived from various industry standards and from observations of additional needs dependent on modeling and simulation, stockpile stewardship, and weapon surety. Each standard was examined and the activities were evaluated based on how they supported the principles, which in turn, uphold the ASCI V&V Program goals. The guidelines were further refined through feedback from code project teams. The final outcome of this process is the following set of guidelines, categorized in three general areas (described in greater detail in the document).

Each ASCI software code project will be expected to develop specific practices that implement these guidelines:

- **Software Verification:** the determination that requirements are accurately and correctly implemented, and that requirements are adequate from the perspective of the intended uses of the software.
- **Software Engineering:** the systematic, disciplined, and quantifiable approach to the development, operation, and support of software, that is, the application of engineering to software.

- **Project Management**: the systematic approach for balancing the project work to be done, resources required, methods to be used, procedures to be followed, schedules to be met, and the way that a project is organized.

As these guidelines and activities were being developed, it became evident this process should be ongoing and evolutionary. Although it is recognized that there are many SQE practices in place within the ASCI program, this document should help develop improvement plans that document the current best practices and recommend further improvement. The path forward should involve the following activities:

- **Develop site-specific specification of practices.** The sites, organizations, and projects that produce software used in the ASCI program are expected to develop specific practices to implement these guidelines.

- **Identify current site-specific state of practices, and mechanisms for improvement.** Sites and development teams will internally identify the state of their practices and establish a strategy for improvement.

- **Improve site-specific practices and document improvement.** Sites and development teams will identify strategies for improvement appropriate to their implementation of these guidelines.

- **Provide feedback for process improvement and future revisions of this document.** As sites and development teams identify and implement best practices, their effectiveness will become evident. Lessons learned as a result will then be incorporated into future revisions of this document.

OVERVIEW

1.1 BACKGROUND

The Department of Energy's Stockpile Stewardship Program (SSP) is designed to ensure confidence in the safety, performance, and reliability of the U.S. nuclear stockpile in the absence of underground testing. The Accelerated Strategic Computing Initiative (ASCI) provides the SSP the needed shift from test-based to science-based methods through high-performance modeling and simulation. A formal, focused verification and validation (V&V) program therefore becomes essential for ensuring the confidence and credibility of ASCI's modeling and simulation activities. Building quality into software products, software quality engineering (SQE), is accomplished through software engineering, software verification, and associated project management activities. These activities contribute to V&V's goals, and must include the systematic measurement, documentation, and demonstration of code capabilities as well as their underlying algorithms, data, and models.

1.2 PURPOSE AND SCOPE

This document describes the goals, principles, and guidelines for SQE activities that support ASCI. As such, it affects all ASCI program elements involved in software development: Advanced Applications Development (not just selected new code development), Materials and Physics Modeling, Problem-Solving Environments, Distance and Distributed Computing and Communication, and Visual Interactive Environment for Weapons Simulation. However, each program element's needs will dictate the level of its adherence to these guidelines (see 1.3 Tailoring Guidelines). The purpose of this document is not to prescribe any specific practices in the form of methods, tools, or measures for implementing these guidelines. It is the responsibility of site-specific organizations and projects to select and appropriately tailor their best practices to achieve stated goals. They should also select, enhance, and subsequently improve their existing SQE practices by implementing these recommended SQE guidelines through a prioritized and graded approach. This document thus serves as a foundation for further process improvements. As a result, ASCI will benefit from improved SQE practices.

Software quality engineering is a part of ASCI's Verification and Validation (V&V) Program. Many of SQE's overall principles and goals are similar to those of V&V; however, the primary focus of SQE is on issues related to software. This document concentrates on providing guidelines for software verification, software engineering, and associated project management activities as they apply to SQE in general, but does not address the broader areas of the overall ASCI V&V Program. How organizations and projects address these overlapping activities by effective practices and work products depends on their own established methods and techniques.

Metrics are important for evaluation. Objective judgment about the effectiveness of SQE practices and possible improvement strategies requires the collection of metrics. It is therefore important to identify and use metrics to contribute to the characterization, understanding, and evaluation of effective practices.

1.3 TAILORING GUIDELINES

This document is to be used as a guideline for implementing the SQE practices within organizations and projects. The selection of specific methods and techniques to implement these practices, and a strategy for improving such practices, should be tailored to effectively meet the guidelines, principles, and goals in this document.

The selection of specific SQE practices, the rigor of their implementation, and the strategy for their improvement will depend on many factors. In a production environment, more rigorous practices should apply than in an environment that is purely exploratory. However, in any environment, care should be taken to identify those practices and associated work products that will be needed to improve the credibility and confidence of software products used for SSP applications.

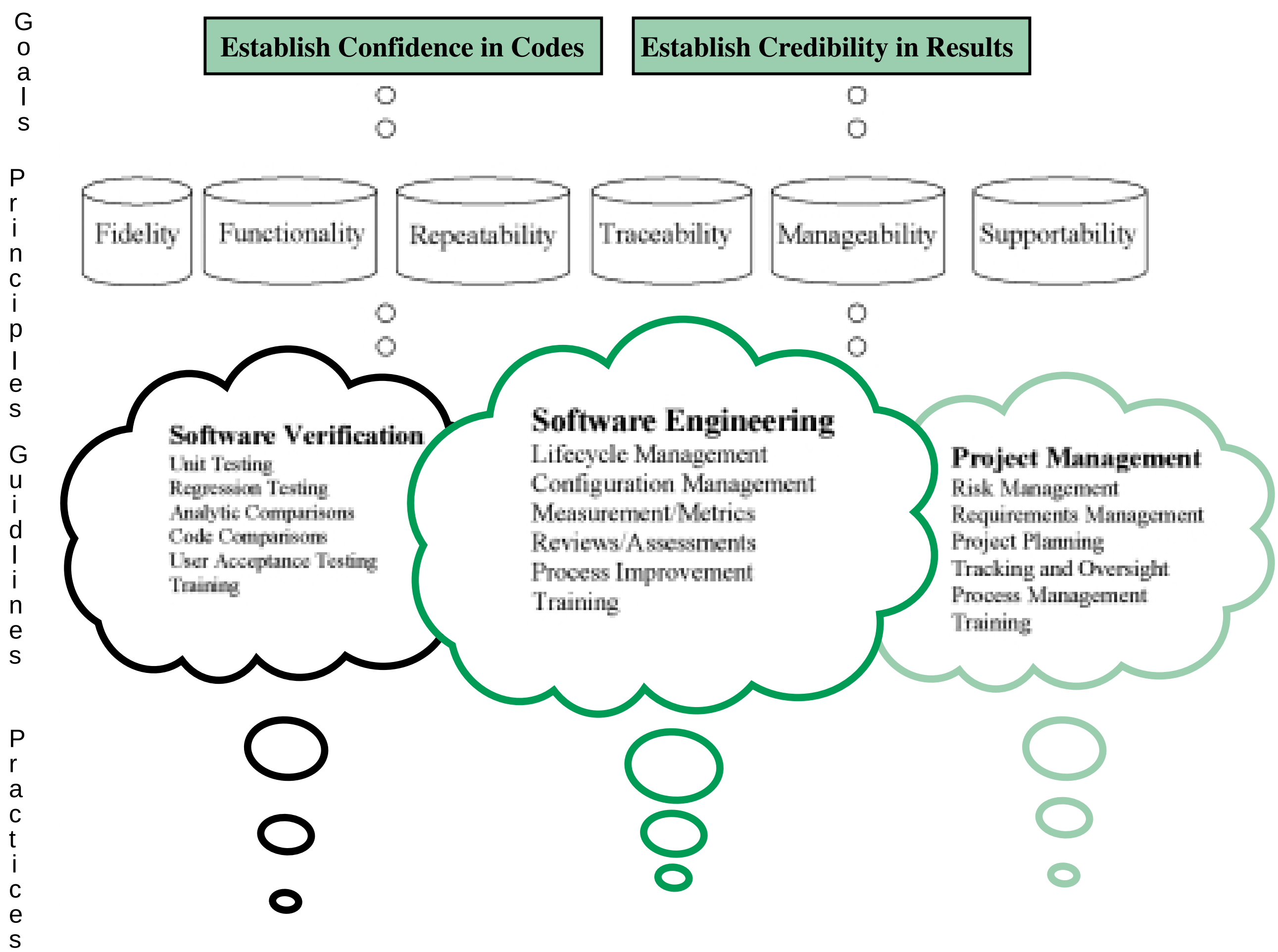
Practices should be tailored based upon a variety of factors such as *size, complexity, cost, formality, degree of impact, visibility, degree of uniqueness, functions performed, criticality, degree of risk, and any additional factors important to the customer or the project team.* When modifying established practices, it is important to assess whether the goals of credibility and confidence are still being met, and to avoid creating practices and work products that do not add any value.

A project that is costly, critical, or risky, should expect to be scrutinized more than a project that does not share those characteristics. For example, exploratory or research and development software, when used for investigative studies, might include a disposable rapid prototype for the exploration of a new algorithm or method. Such software would not be scrutinized at the same level as high-risk software. A graded approach must be applied in evaluating software for purposes of improving software engineering processes.

SECTION 2

STATING GOALS, IDENTIFYING PRINCIPLES, AND SELECTING GUIDELINES FOR SQE

An organization or a project should select and implement SQE practices that support the goals of the ASCI V&V program. Underlying quality principles that support these goals have been identified. Guidelines for SQE activities have been derived to support the principles. Software practices should be adopted by an organization or project in a way that allows the implementation of these SQE guidelines with the appropriate tailoring strategy determined by the individual group. This interdependent relationship of goals, principles, guidelines, and practices is illustrated in the Figure on the following page. These relationships and specific guidelines are discussed in the subsections below.



2.1 GOALS

The goal of the ASCI V&V program is to provide adequate confidence that stockpile certification, using the advanced three-dimensional applications developed by ASCI, has a well-understood range of applicability, and that uncertainties in the results have been analyzed and quantified. Therefore, the V&V program's purpose is to conduct code verification and validation activities intended to produce easily maintained and error-free software packages, and to ensure that the codes give an adequate simulation of physical reality.

This overall goal has two parts as shown in the Figure above:

- Establish confidence in codes, and
- Establish credibility in results.

2.2 PRINCIPLES

Analysis of the ASCI V&V program goals led to identification of several quality principles that drive the selection of SQE practices. The following set of principles is reasonably comprehensive and semantically orthogonal. Precise definitions are not provided, but general attributes of the principles are described.

Fidelity – includes accuracy and correctness of simulation results that enables verification; comparison with analytic and other results; reliability, in the sense of the robustness, likelihood of success, and integrity of results.

Functionality – includes the degree to which the simulation fulfills user requirements, and the capability of the software to accommodate multiple intended applications, as appropriate.

Repeatability – includes the capability to reconstruct simulation results used as the basis of stockpile stewardship decisions; the capability to reproduce those results within the limitations of the existing operational environment; the ability to repeat aggregated modeling and simulation results (this is exemplified when a code is modified or new system software is being utilized); the preservation and retrieval of the essential knowledge (e.g., physics models, input conditions, critical assumptions, and simplifications) from a given application; and the capability to duplicate the project's procedures, use historical data (metrics) to create realistic plans, and to follow the plans.

Traceability – includes the capability to view the progress of a project, and the links and relationships among associated work products; products should include design requirements, codes, inputs, outputs, documentation, key decisions, and conclusions drawn.

Manageability – includes the capability to balance customer requirements (such as schedule, cost, and performance) and customer satisfaction (cost effectiveness, fulfillment of requirements, responsiveness) with programmatic and technical risks; and the capability for management to determine the status, progress, and capabilities of software engineering and other key processes (to characterize, understand, and evaluate them).

Supportability – includes the adequacy of resources and procedures to facilitate modifying and installing software, establishing an operational software baseline, and meeting user-support requirements; the degree to which software and processes are designed for supportability, such as their amenability to change analysis, change implementation, change test, and change release; and the attributes of the software processes, products, and environment that facilitate change. Supportability depends on the design and maturity of code, capability of the resources and systems, and the procedures that are in place.

2.3 GUIDELINES

Guidelines to support the principles are specified below in terms of expected activities in software verification, software engineering, and project management. Specific methods, techniques, and tools are used to implement practices. Practices and associated methods, techniques, and tools are assumed to be site-specific, and implemented according to priorities of the ASCI program in general and of individual development teams. The guidelines for specific activities are explained in greater detail below.

Software Verification – the determination that requirements are accurately and correctly implemented, and that requirements are adequate from the perspective of the intended uses of the software. Additional activities can include: user acceptance testing, documentation of results; capabilities definition and management; testing against both analytical data and other software; and technical reviews. These activities set the stage for establishing confidence in the use of the code.

Software Engineering – the systematic approach to the specification, design, development, test, operation, support, and retirement of software (i.e., the application of engineering to software). Activities should include: life cycle management (requirements, design, construction, test, and support activities in a time-based work flow); configuration management (version management, issue tracking, and release management); measurement of product and process attributes; reviews and assessments of work products and processes; process improvement from a baseline characterization; and training on software engineering activities. The balance among activities, and the relationships with modeling and simulation, verification and validation, and project management depends on many factors including the maturity of the software.

Project Management – the systematic approach for balancing the project work to be done, resources required, methods to be used, procedures to be followed, schedules to be met, and the way that a project is organized. Activities should include: identification, analysis, and mitigation of project risks; controlling requirement changes; planning for project tasks, schedule, and cost; tracking project progress and status; providing oversight of process improvement; and training project personnel in management activities.

2.4 GUIDELINE AREAS

Each guideline area (software verification, software engineering, and project management) has a set of activities and associated practices and work products. Implementation of the practices and development of the work products will be specific to a given organization and project. The table on the following page summarizes these activities and many of their key elements. Section 2.4.1 contains a narrative description of each activity.

2.4.1 SOFTWARE VERIFICATION ACTIVITIES

The activities listed below focus on the software verification activities important to the V&V program in striving towards quality software.

Specific software verification activities should include:

Technical Reviews – the activity of evaluating the technical soundness of work products. This includes analyses to find mismatches or faults between the specification and the design, code, or documentation.

Unit Testing – the activity of testing code units against their requirements, specifications, and design. This activity involves the development and documentation of unit-test drivers and test-case inputs. This activity requires valid work products to be provided by the software developers that clearly and adequately define the requirements, specifications, and design. Unit testing should be developed and performed by the software developers during the development life cycle. It should be traceable and repeatable by an independent V&V team, where it is appropriate to do so.

Regression Testing – the activity of regularly *building* the code and *executing* a series of tests designed to verify that the code works as expected for all computational platforms supported. Minimally, such testing should be done when either the code or operating platform changes. This activity includes the development and maintenance of a regression test suite. This test suite should be designed to exercise as many of the code features as possible. The regression test suite should include integral and unit tests, as appropriate.

GUIDELINE AREA	ACTIVITIES	KEY ELEMENTS
Software Verification	Technical reviews	▲ Technical soundness ▲ Static analysis
	Unit testing	▲ Traceable, repeatable component tests
	Regression testing	▲ Building the code ▲ Executing tests ▲ Feature-based test suite for multiple platforms
	Comparison techniques	▲ Analytic solutions ▲ Other codes' results
	User acceptance testing	▲ Applicability evaluation ▲ Usability evaluation ▲ Code confidence ▲ Results credibility
	Training	▲ Verification methods and techniques
Software Engineering	Life-cycle management	▲ Time-based work flow ▲ Requirements, design, construction, test, support activities
	Configuration management	▲ Version management ▲ Issue tracking ▲ Release management
	Measurements and metrics	▲ Software products ▲ Software processes
	Reviews and assessments	▲ Management reviews ▲ Technical reviews
	Process improvement	▲ Engineering process baseline ▲ Identified improvements ▲ Improvement implementation
	Training	▲ Software practice methods and techniques
Project Management	Risk management	▲ Risk assessment ▲ Risk control
	Requirements management	▲ Gathering, documenting, verifying, managing change to requirements
	Project planning	▲ Statement of work ▲ Constraints and goals ▲ Implementation plan ▲ Resource assessment
	Tracking and oversight	▲ Actual results vs. planned results ▲ Corrective action
	Process management	▲ Process documentation and plans ▲ Technology improvement ▲ Improvement leverage
	Training	▲ Project management methods and techniques

Comparison Techniques – the activity of utilizing additional comparison techniques within the code development team to ensure requirements are being met on a local scale. These activities could include comparing to analytic solutions and to other codes.

User Acceptance Testing – the activity of determining if the work products satisfy the needs of the intended users. This activity should include evaluation of applicability and usability from the end-users' points of view. It is also intended to help build the users' confidence in the codes and their belief in the credibility of the results.

Training – the activity of developing the skills and knowledge of those individuals responsible for software verification activities.

2.4.2 SOFTWARE ENGINEERING ACTIVITIES

An organization's standard software practices define how an organization plans for, manages, builds, tests, and sustains its software products. The organization's standard software practices define the building blocks that are used in different ways to support software projects. There are many software engineering processes, models, and methodologies available.[2]

Specific activities should include:

Life-Cycle Management – the activity of organizing requirements, design, construction, test, and support activities into a time-based work flow. Many life-cycle models exist that could define this activity. The life-cycle model selected and the specific activities of requirements, design, construction, test, and support should have well-defined interfaces with the other software engineering support areas, and should be based on the guiding principles that best achieve the intended applications and overall ASCI V&V Program goals.

Requirements activities should include: methods for gathering requirements from the scientific application modeling domain; analyzing and documenting models that depict required system data, function, and behavior as allocated and traced throughout the application components; verifying that requirements are met in the application design and implementation; and managing any changes to the requirements. Design activities should include repeatable methods for translating requirements information and models (scientific and software) into representations that convey software data structure, architecture, algorithms, and interface features.

Construction activities should include methods that implement a specific software solution of the design, and that can be traced to the design and verified to the specified requirements.

Test activities should include methods to verify the software construction from unit to integrated software components to scientific model application design and requirement specifications, where applicable. These activities overlap with software verification activities to the extent that the activities use similar test suites and results to achieve the required confidence in the software implementation.

Support activities should include methods to manage changes to the implementation of requirements, design, construction, and test work products due to defects that are found, enhancements that are needed,

[2]The Capability Maturity Model™ from the Software Engineering Institute outlines one model from which software engineering processes can be defined. It is not the only model. IEEE and other professional societies as well as many industries and scientific communities provide many alternative software engineering practices and processes for consideration.

or the natural technological evolution within the application domain. Support activities should also include effective interfaces with other software engineering activities such as configuration management for controlling the changes to, and updated releases of, work products.

Configuration Management – the activity of identifying the configuration items in a system, controlling the release and change of those items throughout the system's life cycle, recording and reporting the status of the items and associated change requests, and verifying the completeness and correctness of the items. Configuration management activities are organized into version management, issue tracking, and release management. Version management is the identification and control of the versions of all products, both by individual pieces (e.g., software module) and by appropriate groupings (e.g., set of software modules that constitute an executable program). Issue tracking is the identification and tracking of problems and associated corrective actions, proposed changes for enhancements, and the workflow of activities to accomplish implementation of the change. Release management is the control of product promotion, from development to production use.

Measurements/Metrics – the activity of collecting information for the characterization, understanding, and evaluation of processes and products. Metrics should show how selected site-specific practices satisfy related attributes of specified principles and consequently contribute to meeting the V&V program's goals of confidence in codes and credibility in results. Only metrics that can be demonstrated to assist in meeting project and/or the V&V program's goals should be chosen.[3]

Reviews/Assessments – the activity of examining and evaluating the quality of a process or product. Reviews/assessments should be conducted on work products from all life-cycle phases to catch defects as early as possible. Formality and scope of reviews/assessments, like other activities, should be tailored. Results of the reviews or assessments should be recorded. There are basically two types of reviews:

- **Management reviews** evaluate and communicate status of the project with regard to schedule, cost, and performance; determine whether processes are being followed correctly, particularly with regard to impact on performance, cost, and schedule; and may be internal to the project, or include external personnel and stakeholders.

- **Technical reviews** evaluate technical soundness of work products and processes; include analyses to find mismatches or faults between the specification and the design, code, or documentation; and are conducted by relevant domain experts.

Process Improvement – the activity of baselining the performance of a process through a documented characterization of the actual results achieved by following the process, determining how the process should be improved in comparison with the actual results, and establishing an approach to achieving the improvement.

Training – the activity of developing the skills and knowledge of those individuals responsible for software engineering activities with respect to relevant procedures, tools, and domain knowledge, as they apply to the ASCI program. It also includes training on the use of the developed software products, as well as their domain, scope, and applicability.

[3]There are many known methods for developing a useful measurement program, one of which is the Goal/Question/Metric paradigm. In this method, goals such as the ASCI program goals are established. Then, questions concerned with achieving certain principles, such as those identified in this document, are identified. From these questions, necessary data and derived metrics are gathered in order to answer the questions

2.4.3 PROJECT MANAGEMENT ACTIVITIES

Specific activities should include:

Risk Management – the activity of identifying, addressing, and mitigating sources of risk before they become threats to successful completion of a project.

Risk management elements are:

- risk assessment (identifying, analyzing, and prioritizing);
- risk control (management planning, resolution, and monitoring).

Requirements Management—the activity of capturing, tracking, and controlling requirements, as well as any changes to them. This establishes and maintains a common understanding, between customers and development teams, of the requirements to be addressed by the project. This agreement should be the basis for planning and managing the project.

Project Planning—the activity of establishing a reasonable plan for performing and managing the project; work products should include, but are not limited to, a statement of work, constraints and goals, project plan, project timeline, an assessment of resources that will be needed, and availability of those resources.

Tracking and Oversight—the activity of tracking and reviewing the project accomplishments and results with respect to the project plan, and taking corrective action as necessary based on actual accomplishments and results.

Process Management—the activity related to planning, defining, implementing, monitoring, measuring, and improving processes under project management; and producing process documentation and improvement plans.

Training—the activity of developing the skills and knowledge of those individuals responsible for project management activities.

2.5 RELATIONSHIP OF GUIDELINE AREAS TO EXISTING STANDARDS

There are many sources of standards. All of those reviewed in the preparation of this document are listed in the bibliography. The foundation for the guidelines and activities described here was a distillation and enhancement of the guidelines and activities found in several of the more widely accepted software industry standards. Each standard was examined, and the activities were evaluated for their contribution in support of the principles that in turn support the ASCI V&V Program goals.

Building upon this foundation were additional activities that still support the program goals, but were required to address the SSP needs of modeling and simulation. These activities were extracted from more specialized standards specific to modeling and simulation. In addition, the weapon-surety component of stockpile stewardship, and its implied safety-criticality, necessitated inclusion of further guidelines. These guidelines were extracted from the nuclear facilities standards (as an analog for safety-critical applications), and again evaluated for their support of program goals.

Finally, standards that reflect customer expectations not already captured were consulted. This approach produced the comprehensive, traceable, and defensible set of activities specified in this document, which adhere to the spirit of the reviewed sets of standards.

PATHFORWARD

It is recognized that ASCI development teams are at various stages with regard to the recommended guidelines and activities. A PathForward consists of the following activities:

- **Develop site-specific specification of practices:** Sites, organizations, and projects that produce software used on the ASCI program are expected to develop their own specific practices to appropriately implement the guidelines in this document. Many existing practices may already implement some or all of these guidelines. In other cases, only a partial implementation is appropriate for the identified development team's priorities and tailoring approach. A strategy for appropriate improvement in practices is expected to be developed although the precise form and content of that strategy is left to the individual site.

- **Identify current site-specific state of practices, and mechanism for improvement:** The collection of specified guidelines, practices, and work products can be used to determine the current state of practice at the organizational level. This organizational level can range from a development team, to a programmatic level at a site, or to a programmatic community level (such as the DOE ASCI program). It is expected that sites and development teams will internally identify the state of their practices in order to establish a strategy for improvement.

- **Improve site-specific practices, and document improvement:** Using the identification of the current state and the dependencies for the practices, it is possible to identify steps for practice improvement. It is expected that sites and development teams will identify internal strategies for improvement in keeping with their implementation of this document's guidance.

- **Provide feedback for process improvement and future revisions of this document:** As development teams and sites identify and implement best practices, lessons will be learned about the effectiveness of the practices. These lessons learned should be shared with the ASCI program, especially as they relate to the effectiveness of the goals, principles, and guidelines in this document. Future revisions of this document will incorporate such lessons learned to improve its effectiveness.

There are many SQE practices already in place within the ASCI program. This document is intended to help develop improvement plans that document the current best practices and to recommend further improvements.

It is also recognized that some of the ASCI program elements became subject to these guidelines rather late in the development of this document. Feedback from those elements will be incorporated into future revisions.

APPENDIX A

GLOSSARY

Application: a specific use for which a modeling effort, including the associated codes, is designed, based on the problem solved, the functions or operational capabilities incorporated, and on the hardware system targeted to execute the software.

Change Control: a process which assures that the implementation of modeling and software changes is governed by suitable control measures commensurate with those applied to the original product, and with the scope and nature of the change

Configuration Management: the process of identifying the configuration items in a system; controlling the release and change of those items throughout the system life cycle; recording and reporting the status of the items and associated change requests; and verifying the completeness and correctness of the items. In addition, it should include the complete set of work products involved in a given application, *logically linked*, and controlled as members of that set.

Life Cycle: the organization of requirements, design, construction, test, and support activities into a work flow appropriate for the application. The purpose of a life cycle is to delineate the flow of activities, and to break the modeling, software, and other development processes into steps or phases to facilitate the characterization, understanding, and management of those processes.

Reliability: the probability of failure-free operation of the software for a specified time, in a specified environment.

Requirement: a condition or capability that must be met by a system or system component to satisfy a contract, specification, or other formally imposed document. The set of all requirements forms the basis for system development.

Requirements Management: the capture, tracking, and control of requirements, as well as any changes to them.

Review: a managerial or technical examination of a product or one of its components for the purpose of evaluation of its status, adequacy for intended use, or for detection and remedy of deficiencies.

Software Engineering: the systematic approach to the specification, design, development, test, operation, support, and retirement of software.

Software Validation: the activities of determining the degree to which the *requirements* are adequate, and met, *from the perspective of the intended uses* of the model. (Did you implement and meet the correct requirements?)

Software Verification: the activities of determining that the *requirements* are accurately and correctly implemented. (Did you implement the requirements correctly?)

Work Product: an element of the application that is produced as a result of the development, application of modeling and simulation, and V&V jobs. Some work products might not be delivered to a requestor external to the team.

APPENDIX B

BIBLIOGRAPHY

The following sets of standards comprise those consulted during compilation of this document. Additional standards exist, but those listed were chosen either due to their wide acceptance, or due to their specific applicability to support the ASCI V&V Program goals.

B.1 SOFTWARE STANDARDS

This set of standards provides state of the practice guidance from the software engineering community at large. Application of standards will be tailored for the ASCI mission, and the general mission of the ASCI customer community's applications.

Institute of Electrical and Electronics Engineers. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Standard 610.12-1990. New York, NY, 1990.

Institute of Electrical and Electronics Engineers. *IEEE Standard for Software Test Documentation*. IEEE Standard 829-1998. New York, NY, 1998.

Institute of Electrical and Electronics Engineers. *IEEE Standard for Software Verification and Validation Plans*. IEEE Standard 1012-1998. New York, NY, 1998.

Institute of Electrical and Electronics Engineers. *Supplement to IEEE Standard for Software Verification and Validation Plans: Content Map to IEEE/EIA 12207.1-1997*. IEEE Standard 1012a-1998. New York, NY, 1998.

Institute of Electrical and Electronics Engineers. *IEEE Standard for Software Reviews*. IEEE Standard 1028-1997. New York, NY, 1997.

Institute of Electrical and Electronics Engineers. *IEEE Guide for Software Verification and Validation Plans.* IEEE Standard 1059-1993. New York, NY, 1993.

Institute of Electrical and Electronics Engineers. *IEEE Standard for Application and Management of the Systems Engineering Process.* IEEE Standard 1220-1998. New York, NY, 1998.

Institute of Electrical and Electronics Engineers. *IEEE/EIA Software Life Cycle Processes.* IEEE Standard 12207.0-1996. New York, NY, 1996.

Institute of Electrical and Electronics Engineers. *IEEE/EIA Software Life Cycle Processes – Life Cycle Data.* IEEE Standard 12207.1-1997. New York, NY, 1997.

Institute of Electrical and Electronics Engineers. *IEEE/EIA Software Life Cycle Processes – Implementation Considerations.* IEEE Standard 12207.2-1997. New York, NY, 1997.

B.2 MODELING AND SIMULATION STANDARDS

This set of standards is focused on simulation and modeling.

Defense Modeling and Simulation Office, **http://www.dmso.mil/dmso/index.msql.**

American Institute of Aeronautics and Astronautics, *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations.* AIAA G-077-1998. Reston, VA, 1998.

Joint Accreditation Support Activity (JASA), **http://www.nawcwpns.navy.mil/~jasa/.**

Sandia National Laboratories, Trucano, T., and J. Moya. *Guidelines for Sandia ASCI Verification and Validation Plans – Content and Format.* Version 1.0. Albuquerque, NM, December 1999.

U.S. Department of the Navy, Office of the Secretary. *Verification, Validation and Accreditation (VV&A) of Models and Simulations.* SECNAVINST 5200.40. April 19, 1999.

B.3 NUCLEAR FACILITIES SAFETY STANDARDS

This set of standards reflects safety critical issues.

American National Standards Institute. *American Nuclear Society, Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry.* ANSI/ANS-10.4-1987. New York, NY, 1987.

American Society of Mechanical Engineers. NQA-1-1994. Subpart 2.7. *QA Requirements of Computer Software for Nuclear Facility Applications*, Requirement 11 *Test Control*, Requirement 11S-2 *Supplementary Requirements for Computer Program Testing*. New York, NY, 1994.

Title 40 – Protection of Environment, Part 194 – Criteria for the Certification and Re-Certification of the Waste Isolation Pilot Project, Subpart C – Compliance Certification and Re-Certification, Code of Federal Regulations, **http://www.access.gpo.gov/nara/cfr/waisidx_99/40cfr194_99.html.**

B.4 CUSTOMER EXPECTATION STANDARDS

This set of standards represents various customer expectations from the DOE and DoD communities.

Department of Energy, *ASCI Program Plan*. DOE/DP-99-000010592. Sandia National Laboratories, Albuquerque, NM, January 2000.

DOE Notice 203.1, *Software Quality Assurance*, U.S. Department of Energy, Washington, DC, October 2, 2000. **http://www.explorer.doe.gov:1776/pdfs/doe/doetext/neword/203/n2031.pdf**

Guidelines for Software Measurement, SQAS97-001. **http://cio.doe.gov/sqas/sqas97_1.doc.** April 1997.

http://cio.doe.gov/smp/, DOE Software and Systems Engineering web site, process checklists for Software Configuration Management.

http://cio.doe.gov/smp/, DOE Software and Systems Engineering web site, process checklists for Software Testing.

http://cio.doe.gov/smp/, DOE Software and Systems Engineering web site, process checklists for Software Quality Assurance.

NWC Glossary of Preferred Software Engineering Terminology, SQAS90-001, **http://cio.doe.gov/sqas/sqas90_1.doc.** October 1990.

QC-1 – DOE/AL, Quality Criteria (QC-1). Revision 9. February 5, 1998. **http://prp.lanl.gov:8686/.**

DOE O 414.1A – Quality Assurance, United States Department of Energy, September 19, 1999. **http://prp.lanl.gov:8686/.**

TBP-306 – NWC Technical Business Practice, Software Product Processes. Issue B. July 15, 1999. **http://prp.lanl.gov:8686/**

B.5 ADDITIONAL RESOURCES

TEST DOCUMENTATION RESOURCE MATERIAL

Beizer, B. *Software Testing Techniques*. 2d ed. Van Nostrand Reinhold, New York, NY, 1990.

Black, R. *Managing the Testing Process*. Microsoft Press, Redmond, WA, 1999.

Kaner, C. et al. *Testing Computer Software*. 2d ed. Van Nostrand Reinhold, New York, NY, 1993.

Myers, G. *The Art of Software Testing*. John Wiley and Sons, New York, NY, 1979.

MISCELLANEOUS RESOURCE MATERIAL

Ambrosiano, John, and Maysa-Maria Peterson. Research Software Development Based on Exploratory Workflows: The Exploratory Process Model (ExP). LA-UR-00-3697. Los Alamos National Laboratory Los Alamos, NM, 2000.

Boehm, Barry W. "A Spiral Model of Software Development and Enhancement." *Computer,* Volume 21, Issue 5, May 1988: 61-72.

Fowler, M. et al. *UML Distilled*. Addison-Wesley Publishing Co, Reading, MA, 1997.

Gilb, Tom, and Susannah Finzi (eds.). *Principles of Software Engineering Management*. Wokingham, England: Addison-Wesley Publishing Co, Reading, MA, 1988.

Hodges, A., G. Froehlich, M. Pilch, and D. Peercy. *Sandia National Laboratories ASCI Applications V&V Software Quality Engineering Guidelines: Practices and Work Products*, Revision 1.10. Albuquerque, NM, Sandia National Laboratories, September 2000.

Humphrey, Watts. *Introduction to the Personal Software Process*. Addison-Wesley Publishing Co, Reading, MA, 1997.

Humphrey, Watts S. *Managing the Software Process*. Software Engineering Institute Series in Software Engineering. Addison-Wesley Publishing Co, Reading, MA, 1989.

McConnell, Steve. *Code Complete*. Microsoft Press, Redmond, WA, 1993.

McConnell, Steve. *Rapid Development*. Microsoft Press, Redmond, WA, 1996.

Naur, P., and B. Randell (eds.). Software Engineering: A Report on a Conference sponsored by the NATO Science Committee. NATO, Garmisch, Germany, 1969.

Paulk, Mark C. et al. *The Capability Maturity Model: Guidelines for Improving the Software Process.* SEI Series in Software Engineering, Carnegie Mellon University, Pittsburgh, PA, 1994.

Pressman, Roger S. *Software Engineering: A Practitioner's Approach.* 3d ed. New York: McGraw-Hill, 1992.

Sommerville, Ian. *Software Engineering*. 4h ed. Addison-Wesley Publishing Co, Reading, MA, 1992.

Strategic Computing and Simulation Validation and Verification Program – Program Plan, Version 2.5, April 1998, **http://www-irn.sandia.gov/organization/div9000/ctr9100/dept9105/VnVhome/vnvframes.html** and then click on "V&V Plans."

Wiegers, K. *Software Process Improvement: Eight Traps to Avoid.* Crosstalk, **http://www.stsc.hill.af.mil/CrossTalk/1998/sep/wiegers.asp,** September 1998.

DISTRIBUTION

Sandia National Laboratories (153)
1515 Eubank Ave SE
P.O. Box 5800
Albuquerque, NM 87123

NAME	MAIL STOP	ORG	NO.
Abeyta, Henry	MS-0453	02101	1
Ang, James	MS-0321	09224	1
Aragon, Kathryn	MS-0827	09131	1
Attaway, Steve	MS-0847	09121	1
Baca, Stephen	MS-0957	14111	1
Baca, Thomas	MS-0557	09125	1
Bickel, Thomas	MS-0841	09100	1
Blackledge, Michael	MS-0638	12326	1
Blanford, Mark	MS-0847	09121	1
Boucheron, Edward	MS-0819	09231	1
Budge, Kent	MS-0819	09231	1
Burns, Shawn	MS-0835	09111	1
Byle, Kathleen	MS-0660	09519	1
Camp, William	MS-0321	09200	1
Carlson, David	MS-0428	12300	1
Chavez, Patrick	MS-0820	09232	1
Cochran, Robert	MS-0835	09111	1
Crawford, Dona	MS-9003	09900	1
D'Antonio, Perry	MS-0490	09713	1
Davis, David	MS-1135	09134	1
Detry, Ronald	MS-0421	09800	1
Dickinson, Mark	MS-0470	09821	1
Diegert, Kathleen	MS-0829	12335	1
Easterling, Robert	MS-0419	09800	1
Eaton, Donna	MS-0660	09519	1
Edwards, H. Carter	MS-0826	09131	1
Ellis, Larry	MS-1140	06502	1
Ellis, Molly	MS-0660	09519	1
Ernest, Martha	MS-0806	09336	1
Fernandez, Joseph	MS-0828	09140	1
Forsythe, Christie	MS-0826	09143	1
Froehlich, Gary	MS-1137	06535	5
Garber, Reeta A.	MS-0828	09140	1
Griffith, Richard	MS-0836	09117	1
Hagengruber, Roger	MS-1231	05000	1
Hale, Arthur	MS-0321	09220	1
Hannah, Michael	MS-0807	09338	1
Harris, Pamela	MS-1137	06535	1
Hartwig, Ronald	MS-0427	02100	1
Hassan, Basil	MS-0835	09141	1
Heermann, Philip	MS-0318	09227	1
Heffelfinger, Grant	MS-0316	09235	1
Hermina, Wahid	MS-0826	09113	1
Hertel, Eugene	MS-0836	09116	1
Hiebert-Dodd, Kathie	MS-1137	06535	1
Hodges, Ann	MS-1137	06534	5
Hogan, Roy	MS-0836	09117	1
Hunter, Thomas	MS-0151	09000	1
Johannes, Justine	MS-0834	09114	1
Johnson, Victor	MS-0428	12301	1
Jones, Todd	MS-0405	12333	1
Kelly, John E	MS-0143	09904	1
Kempka, Steven	MS-0835	09141	1
Key, Samuel	MS-0847	09121	1
Kiefer, Mark	MS-1152	01642	1
Knirk, Dwayne	MS-0638	12326	1
Koszykowski, Michael	MS-9217	08950	1
Lawrence, R. J.	MS-1186	01674	1
Leland, Robert	MS-0847	09226	1
Lober, Randall	MS-0835	09111	1
Lorence, Leonard	MS-1179	15341	1
Martinez, David	MS-0847	09124	1
May, Rodney	MS-0847	09126	1
McCornack, Marjorie	MS-1138	06534	1
McGlaun, J. Michael	MS-0827	09140	1
McWherter-Payne, Mary	MS-0825	09115	1
Merillat, Paul	MS-0629	09500	1
Meza, Juan	MS-9217	08950	10

Mitchiner, John	MS-1137	06534	1
Moen, Christopher	MS-9042	08728	1
Moffatt, William	MS-0481	02114	1
Morgan, Harold	MS-0847	09123	1
Moya, Jaime	MS-0828	09132	1
Oberkampf, William	MS-0828	09133	1
Paez, Thomas	MS-0828	09133	1
Pasik, Michael	MS-1152	01642	1
Peercy, David	MS-0638	12326	5
Peery, James	MS-0835	09142	1
Peterson, Carl	MS-0836	09104	1
Pilch, Martin	MS-0828	09133	10
Pundit, Neil	MS-1110	09223	1
Rao, Rekha	MS-0827	09114	1
Ratzel, Arthur	MS-0834	09112	1
Reese, Garth	MS-0847	09121	1
Rice, James	MS-0953	02500	1
Romero, Vicente	MS-0828	09133	1
Rosenthal, Mark	MS-0481	02114	1
Rottler, J. Stephen	MS-0429	02100	1
Rutledge, Walter	MS-0825	09115	1
Sena, Patrick	MS-0427	02104	1
Sjaardema, Gregory	MS-0847	09131	1
Stichman, John	MS-0457	02000	1
Summers, Randall	MS-0819	09231	1
Tedeschi, William	MS-0479	02113	1
Tegnelia, James	MS-1221	15000	1
Tieszen, Sheldon	MS-0836	09116	1
Trucano, Timothy	MS-0819	09211	5
Vahle, Michael	MS-0801	09300	1
Varnado, Samuel	MS-1140	06500	1
Washington, Kenneth	MS-9003	08900	1
Witkowski, Walter	MS-0828	09133	1
Wix, Steven	MS-0525	01734	1
Wojtkiewicz, Steven	MS-0847	09124	1
Wolfe, Walter	MS-0835	09111	1
Womble, David	MS-1110	09214	1
Wong, Michae1	MS-0819	09231	1
Woodard, Joan	MS-0102	00002	1
Yarrington, Paul	MS-0820	09230	1
Yonas, Gerold	MS-0839	16000	1
Young, Eunice	MS-1138	06532	1
Zepper, John	MS-0826	09143	1

Central Technical Files	MS-9018	08945-1	1
Technical Library	MS-0899	09616	2
Review & Approval Desk for DOE/OSTI	MS-0612	09612	1

University of California (75)
Lawrence Livermore National Laboratory
7000 East Ave.
P.O. Box 808
Livermore, CA 94550

NAME	MAIL STOP	ORG	NO.
Adams, Thomas	L-095	B/DNT	1
Anastasio, Michael	L-160	DNT	1
Carlson, Gary	L-096	B/DNT	1
Chidester, Steven	L-125	W/DNT	1
Christensen, Randy	L-160	ASCI/DNT	1
Cook, Linnea	L-098	Computations	1
Dube, Evelyn	L-095	B/DNT	1
Goodwin, Bruce	L-170	B/DNT	1
Hoover, Carol	L-125	W/DNT	1
Hopkins, Harvey	L-125	W/DNT	1
Klein, Richard	L-023	A/DNT	10
Lee, Anthony	L-125	W/DNT	1
Logan, Roger	L-125	W/DNT	10
Louis, Steven	L-060	ASCI/VIEWS	1
Mara, Glenn	L-125	W/DNT	1
Michels, Ted	L-066	Computations	1
Miller, Doug	L-013	A/DNT	1
Miller, Patrick	L-013	A/DNT	1
Mish, Kyran	L-125	W/DNT	1

Nitta, Cynthia	L-096	B/DNT	10
Nowak, David	L-160	ASCI/DNT	20
Persons, Warren	L-501	Computations	1
Quinn, Terri	L-061	ASCI/VIEWS	1
Rathkopf, Jim	L-013	A/DNT	1
Sam, David	L-125	W/DNT	1
Sharp, Richard	L-096	B/DNT	1
Slavik, Todd	L-125	W/DNT	1
Storch, Nancy	L-303	Computations	1
Ward, Richard	L-013	A/DNT	1

Los Alamos National Laboratory (66)
Mail Station 5000
P.O. Box 1663
Los Alamos, NM 87545

NAME	MAIL STOP	ORG	NO.
Addessio, Frank	MS-B216	T-3	1
Bishop, Alan	MS-B210	T-DO	1
Brown, Jeff	MS-B272	CCN-DO	1
Cox, Lawrence	MS-F663	X-5	1
Clark, Gary	MS-B295	CCN-12	1
Day, Robert	MS-P915	DX-DO	1
Dotson, Paul	MS-F652	NW-SC	1
Hanson, Earle	MS-P945	ESA-DO	1
Koch, Ken	MS-F652	NW-SC	25
Lagrange, Jeanette	MS-B295	CCN-12	1
Lee, Stephen	MS-F652	NW-SC	1
Lemons, Ross	MS-G754	MST-DO	1
Mangeng, Carolyn	MS-A105	ALDNW	1
McCoy, Don	MS-F652	NW-SC	1
McKinney, Gregg	MS-F663	X-5	1
Mercer-Smith, Jas	MS-A105	ALDNW	1
Miller, Charlie	MS-F652	NW-SC	1
Morrison, John	MS-B260	CIC-DO	1
Peterson-Schnell, Maysa	MS-B295	CCN-12	15
Post, Doug	MS-B218	X-DO	1
Rider, William	MS-D413	CCS-2	1
Sattelberger, Al	MS-J515	CST-DO	1
Seed, Tom	MS-665	X-5	1
Thompson, Buck	MS-B297	CCS-DO	1
Tomlinson, Bob	MS-B260	CCN-DO	1
Weisheit, Jon	MS-B218	X-DO	1
White, Andy	MS-B297	CCS-DO	1
Witt, Larry	MS-F630	NS-SS	1

Department of Energy (11)
1000 Independence Ave SW
Washington, DC 20585

NAME	MAIL STOP	ORG	NO.
Reed, William	DP-14		10
Thomas, Robert	DP-14		1

SAND 2001-0253P

Issued by Sandia National Laboratories, a Department of Energy Laboratory operated by Sandia Corporation, a Lockheed Martin Company, under contract DE-AC04-94AL85000.

Available to United States Department of Energy employees and contractors from the Office of Scientific and Technical Information, U.S. Department of Energy, Post Office Box 62, Oak Ridge, TN 37831. Telephone: 423-576-8401. Telefax:423-576-2865. Internet address: reports@adonis.osti.gov.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161. Telephone: 703-487-4650. Telefax: 703-321-8547.

Lawrence Livermore National Laboratory

Los Alamos
NATIONAL LABORATORY

Sandia National Laboratories